

FAULT DETECTION IN METALLIC PRODUCT USING MACHINE LEARNING TECHNIQUES



By
Tasnuva Jahan Nuva
20MME009F

A thesis submitted in partial fulfillment of the requirements for the degree of
MASTER of SCIENCE in Mechanical Engineering

Department of Mechanical Engineering
CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY

2024

APPROVAL

The thesis titled “**Fault detection in Metallic Product Using Machine Learning Techniques**” submitted by Tasnuva Jahan Nuva (ID:20MME009F, Session: 2020-2021) has been accepted as satisfactory by the Dissertation Committee for the partial fulfillment of the requirement for the degree of Master of Science (M.Sc.) in Mechanical Engineering, dated on 31 October,2024.

Dissertation Committee

Dr. Sajal Chandra Banik
Professor,
Department of Mechanical Engineering, CUET.

Chairman
(Supervisor)

Dr. Mohammad Mizanur Rahman
Head,
Department of Mechanical Engineering, CUET.

Member
(Ex Officio)

Dr. Jamal Uddin Ahamed,
Professor,
Department of Mechanical Engineering, CUET.

Member

Dr. Kazi Afzalur Rahman
Professor,
Department of Mechanical Engineering, CUET.

Member

Dr. Md. Rokunuzzaman
Professor,
Department of Mechanical Engineering, RUET.

Member
External

Declaration

I hereby declare that the work in this thesis has not been previously submitted to meet the requirements for an award at this or any other higher education institution. To the best of my knowledge and belief, the thesis contains no material previously published or written by another person except where due reference is cited. Furthermore, the Thesis compiles with PLAGIARISM and ACADEMIC INTEGRITY regulation of CUET.

Tasnuva Jahan Nuva

20MME009F

Department of Mechanical Engineering

Chittagong University of Engineering & Technology (CUET)

Copyright © Tasnuva Jahan Nuva, 2024

This work may not be copied without permission of the author or Chittagong University of Engineering & Technology

List of Publications

Tasnuva Jahan Nuva, Sajal Chandra Banik, Nahid Hasan and S. M. Fahim Faisal, “Detecting Industrial Product Defects: An Unsupervised Machine Learning Approach for Anomaly Detection” 2nd International Conference on Mechanical, Manufacturing and Process Engineering (ICMMPE – 2024), Paper ID: 79,2024.

Approval by the Supervisor

This is to certify that Tasnuva Jahan Nuva (ID:20MME009F) has carried out this work under my supervision and that she has fulfilled the relevant Academic Ordinance of the Chittagong University of Engineering and Technology so that she is qualified to submit the following thesis in application for the degree of MASTER of SCIENCE in Mechanical Engineering. Furthermore, the Thesis compiles with PLAGIARISM and ACADEMIC INTEGRITY regulation of CUET.

Dr. Sajal Chandra Banik

Professor

Department of Mechanical Engineering
Chittagong University of Engineering & Technology

Abstract

In the realm of manufacturing, ensuring product quality is critical for maintaining operational efficiency and meeting customer satisfaction standards. Defect detection and anomaly identification are key elements of quality control processes. This thesis proposes a machine learning-based approach to the detection and segmentation of faults in metallic products, utilizing three advanced techniques: Mini Batch Dictionary Learning with Sparse Coder, a custom U-Net model, and DeepLabV3+ algorithms.

The research focuses on six distinct metallic objects—cable, grid, metal nut, screw, transistor, and zipper—using the MVTec AD anomaly detection dataset, which includes both defective and defect-free images. For unsupervised anomaly detection, the Mini Batch Dictionary Learning method is employed, demonstrating high precision with an Average Precision (AP) score of 0.976 across the selected metallic objects. Additionally, a custom U-Net model is developed and trained for fault segmentation, providing detailed pixel-level detection of defects on metallic surfaces. The U-Net model achieved high accuracy levels, ranging from 87.92% to 99.59% for different object types, indicating its strong applicability in industrial environments. Finally, DeepLabV3+ model is incorporated to improve segmentation accuracy through the enhancement of defect detection and classification capabilities. The results of this study validate the effectiveness of machine learning algorithms in automating Defect detection in industrial products, thereby reducing human error, minimizing waste, and improving overall production quality. A comparative analysis demonstrates the competitiveness of the proposed approach against alternative algorithms. Future research should focus on refining the segmentation of faulty regions and exploring additional performance metrics. This study contributes significantly to the field of industrial anomaly detection, offering valuable insights into enhancing quality control procedures within industrial settings.

বিমূর্ত

উৎপাদনের দক্ষতা বজায় রেখে গ্রাহকের সন্তুষ্টি অর্জনের জন্য উৎপাদন শিল্পে পণ্যের গুণগত মান নিশ্চিত করা অত্যন্ত গুরুত্বপূর্ণ। ত্রুটি সনাক্তকরণ এবং অস্বাভাবিকতা চিহ্নিতকরণ গুণগত নিয়ন্ত্রণ প্রক্রিয়ার মূল উপাদান। এই গবেষণাপত্রে ধাতব পণ্যের (ক্যাবল, গ্রিড, মেটাল নাট, স্ক্রু, ট্রানজিস্টর, জিপার) ত্রুটি সনাক্তকরণ ও শ্রেণীবিভাজনের জন্য তিনটি উন্নত মেশিন লার্নিং প্রযুক্তি—স্পার্স কোডার সহ মিনি ব্যাচ অভিধান শেখার, কাস্টম ইউ-নেট মডেল এবং ডিপল্যায V3+—ব্যবহারের প্রস্তাব দেওয়া হয়েছে। মিনি ব্যাচ ডিকশনারি লার্নিং পদ্ধতি ব্যবহার করে অযাচিত ত্রুটি সনাক্ত করা হয়েছে, যা ধাতব বস্তুগুলোর ক্ষেত্রে ০.৯৭৬ এর সর্বসাধারণ নির্ভুলতা (AP) স্কোর সহ অত্যন্ত সঠিকভাবে ত্রুটি সনাক্ত করতে সক্ষম। অতিরিক্তভাবে, একটি কাস্টম ইউ-নেট মডেল উন্নয়ন করা হয়েছে। এই মডেলটি প্রশিক্ষণ করা হয়েছে ধাতব পণ্যের পৃষ্ঠের ত্রুটির পিক্সেল-লেভেল সনাক্তকরণের জন্য। ইউ-নেট মডেলটি বিভিন্ন ধাতব বস্তুর ক্ষেত্রে ৮৭.৯২% থেকে ৯৯.৫৯% পর্যন্ত উচ্চমানের নির্ভুলতা অর্জন করেছে, যা বিশেষ করে ইলেকট্রনিক্স এবং অটোমোটিভ শিল্পে এর কার্যকারিতা প্রমাণ করে। পরিশেষে, ডিপল্যায V3+ মডেল অর্ন্তভুক্ত করা হয়েছে, যা সেগমেন্টেশন নির্ভুলতাকে আরও উন্নত করে, ত্রুটি সনাক্তকরণ ও শ্রেণীবিভাজন ক্ষমতা বাড়িয়েছে। এই গবেষণার ফলাফল প্রমাণ করে যে, মেশিন লার্নিং এলগরিদমগুলো শিল্পে পণ্যের ত্রুটি সনাক্তকরণের স্বয়ংক্রিয়তা অনায়াসে সক্ষম করে, যা পণ্যের ত্রুটি নির্ধারণে মানুষের দ্বারা সৃষ্ট ভুল হ্রাস করে, বর্জ্য কমায় ও সামগ্রিক উৎপাদন গুণগত মান উন্নত করে। একটি তুলনামূলক বিশ্লেষণ প্রস্তাবিত পদ্ধতিটি অন্যান্য এলগরিদমের সাথে প্রতিযোগিতামূলক প্রমাণিত হয়েছে। ভবিষ্যৎ গবেষণার জন্য ত্রুটিযুক্ত অঞ্চলের আরও নিখুঁত বিভাজন এবং পরিমাপের জন্য অতিরিক্ত কর্মক্ষমতার সূচকগুলি অন্বেষণ করার প্রস্তাব দেওয়া হয়েছে। এই গবেষণা উৎপাদন শিল্পে অ্যানোমালি সনাক্তকরণ ক্ষেত্রে গুরুত্বপূর্ণ অবদান রাখে এবং শিল্পক্ষেত্রে গুণগত নিয়ন্ত্রণ প্রক্রিয়াগুলোর উন্নতির জন্য মূল্যবান দিক নির্দেশনা প্রদান করে।

Acknowledgment

Above all, I would like to express my gratitude to my supervisor, **Prof. Dr. Sajal Chandra Banik**, Professor, Dept. of ME, CUET for accepting me into his research group and also express my heartfelt thanks to him for his guidance, encouragement, and continuous support during my graduate studies. His enthusiasm for teaching and research offered me challenging opportunities to expand my scientific knowledge and my growing interest in the world of Industrial Engineering specifically in quality control and management.

Table of Contents

APPROVAL.....	ii
Declaration	iii
List of Publications	iv
Approval by the Supervisor	v
Abstract	vi
বিমূর্ত	vii
Acknowledgment	viii
LIST OF FIGURES	xii
LIST OF TABLES	xvii
LIST OF SYMBOLS AND ABBREVIATIONS	xviii
Chapter 01	1
Introduction	1
1.1 Background and motivation	1
1.2 Thesis objectives	5
1.3 Context of defect detection on metallic product	5
1.4 Context of segmentation of metallic product surface defects using Custom U-Net model.....	6
1.5 Context of segmentation of metallic product surface defects using DeepLabv3+ model	6
1.6 Conclusion	7
Chapter 02	9
Literature Review	9
2.1 Introduction.....	9
2.2 Previous study about defect detection and segmentation of defect in metallic products using various ML algorithms	9
2.3 Conclusion	23
Chapter 03	24
Methodology	24
3.1 Introduction.....	24
3.1.1 Image Processing Algorithms Employed.....	24
3.1.2 The reasons behind choosing the ML algorithms are such as:.....	26
3.2 Defect detection using Mini Batch Dictionary Learning Algorithm.....	31
3.2.1 Data Collection	32

3.2.2	Mini Batch Dictionary Learning Algorithm	32
3.2.3	Sparse Coder	33
3.2.4	VGG16	33
3.2.4.1	VGG16 Architecture	34
3.2.5	Methodology of Anomaly Detection	36
3.3	Defect segmentation on metallic products using Custom U-net model	41
3.3.1	Data Processing	41
3.3.2	Model Training	42
3.3.3	Segmentation Process	46
3.4	Defect segmentation on metallic products using DeepLabV3+ model.....	47
3.5	Conclusion	49
Chapter 04		50
Results and Discussion.....		50
4.1	Introduction.....	50
4.2	Results obtained for Defect detection on metallic products using Mini Batch Dictionary Learning with Sparse Coder.....	50
4.2.1	Results obtained for anomaly detection (unsupervised approach) for cable using Mini Batch Dictionary Learning with Sparse Coder model.....	51
4.2.2	Results obtained for anomaly detection (unsupervised approach) for Grid using Mini Batch Dictionary Learning with Sparse Coder model.....	52
4.2.3	Results obtained for anomaly detection (unsupervised approach) for Metal Nut using Mini Batch Dictionary Learning with Sparse Coder model.....	54
4.2.4	Results obtained for anomaly detection (unsupervised approach) for Screw using Mini Batch Dictionary Learning with Sparse Coder model.....	55
4.2.5	Results obtained for Defect detection and segmentation for Transistor using Mini Batch Dictionary Learning with Sparse Coder model	57
4.2.6	Results obtained for Defect detection and segmentation for Zipper using Mini Batch Dictionary Learning with Sparse Coder model	58
4.3	Results obtained for Defect detection using segmentation on metallic products using Custom U-Net model	62
4.3.1	Results obtained for Defect detection using segmentation for cable using Custom U-Net model.....	62
4.3.2	Results obtained for Defect detection using segmentation for Grid using Custom U-Net model.....	68
4.3.3	Results obtained for Defect detection using segmentation for Metal nut using Custom U-Net model	72

4.3.4 Results obtained for Defect detection using segmentation for Screw using Custom U-Net model.....	76
4.3.5 Results obtained for Defect detection using segmentation for Transistor using Custom U-Net model	80
4.3.6 Results obtained for Defect detection using segmentation for Zipper using Custom U-Net model.....	85
4.4 Results obtained for Defect detection using segmentation on metallic products using DeeplabV3+ model	89
4.4.1 Results obtained for Defect detection and segmentation for cable using DeeplabV3+ model	90
4.4.2 Results obtained for Defect detection and segmentation for Grid using DeeplabV3+ model	95
4.4.3 Results obtained for Defect detection and segmentation for Metal nut using DeeplabV3+ model	101
4.4.4 Results obtained for Defect detection and segmentation for Screw using DeeplabV3+ model	104
4.4.5 Results obtained for Defect detection and segmentation for Transistor using DeeplabV3+ model	108
4.4.6 Results obtained for Defect detection and segmentation for Zipper using DeeplabV3+ model	112
4.5 Conclusion	116
Chapter 05	118
Conclusion	118
5.1 Introduction.....	118
5.2 Summary of findings.....	118
5.3 Contribution of This Research	122
5.4 Future research directions	123
5.5 Better performance of these proposed models	124
5.6 Comparison to Existing Methods.....	125
5.7 Concluding remarks	126
References	127

LIST OF FIGURES

Figure 1. 1: Cable.....	4
Figure 1. 2: Grid.....	4
Figure 1. 3: Metal nut.....	4
Figure 1. 4: Screw	4
Figure 1. 5: Transistor.....	5
Figure 1. 6: Zipper.....	5
Figure 3. 1: VGG 16 layer.....	35
Figure 3. 2: Methodology for training purpose.....	38
Figure 3. 3: Methodology applied for testing purpose.....	40
Figure 3. 4: Architecture of U-Net model.....	43
Figure 3. 5: Training flowchart of custom U-Net model	44
Figure 3. 6: Segmentation flowchart of Custom U-Net model	45
Figure 3. 7: Architecture of DeepLabV+ Model[56]	48
Figure 4. 1: Missing Wire defect of cable.....	51
Figure 4. 2: Missing cable defect of cable	51
Figure 4. 3: Cut cable inner insulation defect of cable	51
Figure 4. 4: Combined defect of cable	52
Figure 4. 5: Bent defect of Grid	52
Figure 4. 6: Broken defect of Grid	52
Figure 4. 7: Glue defect of Grid	53
Figure 4. 8: Good Image of Grid.....	53
Figure 4. 9: Metal_contamination defect of Grid.....	53
Figure 4. 10: Thread defect of Grid	53
Figure 4. 11: Bent defect of metal nut.....	54
Figure 4. 12: Good image of metal nut	54
Figure 4. 13: Color defect of metal nut.....	54

Figure 4. 14: Flip defect of metal nut.....	55
Figure 4. 15: Manipulated front defect of screw	55
Figure 4. 16: Scratch_head defect of screw	56
Figure 4. 17: Scratch_neck defect of screw	56
Figure 4. 18: Thread side defect of screw	56
Figure 4. 19: Thread_top defect of screw	57
Figure 4. 20: bent_lead defect of transistor.....	57
Figure 4. 21: Cut_lead defect of transistor.....	57
Figure 4. 22: Misplaced defect of transistor.....	58
Figure 4. 23: Damaged_case defect of transistor	58
Figure 4. 24: Broken teeth defect of zipper.....	58
Figure 4. 25: Combined defect of zipper	59
Figure 4. 26: Fabric border defect of zipper	59
Figure 4. 27: Fabric interior defect of zipper	59
Figure 4. 28: Good image of zipper	59
Figure 4. 29: Rough defect of zipper	60
Figure 4. 30: Split teeth defect of zipper.....	60
Figure 4. 31: Learning curve for cable using Custom U-Net model.....	63
Figure 4. 32: Accuracy of the trained U-Net model for cable	64
Figure 4. 33: Predicted mask for the trained U-Net model for cable.....	65
Figure 4. 34: Bent wire defect of cable	66
Figure 4. 35: Cable swap defect of cable	66
Figure 4. 36: Combined defect of cable	66
Figure 4. 37: Cut inner insulation defect of cable.....	67
Figure 4. 38: Cut outer insulation defect of cable.....	67
Figure 4. 39: Missing cable defect of cable	67
Figure 4. 40: Poke insulation defect of cable.....	67
Figure 4. 41: Learning curve for Grid using Custom U-Net model.....	68
Figure 4. 42: Accuracy of the trained model for Grid.....	69
Figure 4. 43: Predicted mask for the trained model for Grid	70

Figure 4. 44: Bent defect of Grid	71
Figure 4. 45: Broken defect of Grid	71
Figure 4. 46: Glue defect of grid.....	71
Figure 4. 47: Learning curve for Metal nut using Custom U-Net model.....	72
Figure 4. 48: Accuracy of the trained model for Metal nut	73
Figure 4. 49: Predicted mask for the trained model for Metal nut.....	74
Figure 4. 50: Bent defect of metal nut.....	75
Figure 4. 51: Color defect of metal nut.....	75
Figure 4. 52: Flip defect of metal nut.....	75
Figure 4. 53: Scratch defect of metal nut	76
Figure 4. 54: Learning curve for screw using Custom U-Net model.....	77
Figure 4. 55: Accuracy of the trained model for Screw	78
Figure 4. 56: Predicted mask for the trained model for Screw	79
Figure 4. 57: Manipulated front defect of screw.....	79
Figure 4. 58: Scratch head defect of screw	79
Figure 4. 59: Scratch neck defect of screw	80
Figure 4. 60: Thread side defect of screw	80
Figure 4. 61: Thread top defect of screw	80
Figure 4. 62: Learning curve for transistor using Custom U-Net model	81
Figure 4. 63: Accuracy of the trained model for Transistor.....	82
Figure 4. 64: Predicted mask for the trained model for Transistor	83
Figure 4. 65: Damaged case defect of transistor.....	84
Figure 4. 66: Cut lead defect of transistor.....	84
Figure 4. 67: Bent lead defect of transistor.....	84
Figure 4. 68: Misplaced defect of transistor.....	85
Figure 4. 69: Learning curve for Zipper using Custom U-Net model	85
Figure 4. 70: Accuracy of the trained model for Zipper	86
Figure 4. 71: Predicted mask for the trained model for Zipper.....	87
Figure 4. 72: Fabric border defect of zipper	88
Figure 4. 73: Fabric border interior of zipper	88

Figure 4. 74: Rough defect of zipper	88
Figure 4. 75: Split teeth defect of zipper	89
Figure 4. 76: Squeezed teeth defect of zipper	89
Figure 4. 77: Learning curve for cable using DeeplabV3+ model.....	90
Figure 4. 78: Accuracy of the trained model for cable	92
Figure 4. 79: Predicted mask for the trained model for cable	93
Figure 4. 80: Poke insulation defect of Cable	94
Figure 4. 81: Cut inner insulation defect of Cable	94
Figure 4. 82: Missing wire defect of Cable.....	94
Figure 4. 83: Combined defect of Cable	94
Figure 4. 84: Cable swap defect of Cable	95
Figure 4. 85: Bent wire defect of Cable	95
Figure 4. 86: Learning curve for grid using DeeplabV3+ model.....	96
Figure 4. 87: Accuracy of the trained model for Grid.....	97
Figure 4. 88: Predicted mask for the trained model for Grid	99
Figure 4. 89: Bent defect of grid	99
Figure 4. 90: Broken defect of grid	100
Figure 4. 91: Glue defect of grid.....	100
Figure 4. 92: Metal contamination defect of grid	100
Figure 4. 93: Thread defect of grid	100
Figure 4. 94: Learning curve for metal nut using DeeplabV3+ model	101
Figure 4. 95: Accuracy of the trained model for Metal nut	102
Figure 4. 96: Predicted mask for the trained model for Metal nut.....	103
Figure 4. 97: Bent defect of metal nut.....	104
Figure 4. 98: Color defect of metal nut.....	104
Figure 4. 99: Flip defect of metal nut.....	104
Figure 4. 100: Learning curve for Screw using DeeplabV3+ model	105
Figure 4. 101: Accuracy of the trained model for Screw	106
Figure 4. 102: Predicted mask for the trained model for Screw	107
Figure 4. 103: Manipulated front defect of screw.....	107

Figure 4. 104: Scratch head defect of screw	108
Figure 4. 105: Thread side defect of screw	108
Figure 4. 106: Learning curve for transistor using DeeplabV3+ model	109
Figure 4. 107: Accuracy of the trained model for transistor	109
Figure 4. 108: Predicted mask for the trained model for transistor	110
Figure 4. 109: Bent lead defect of transistor	111
Figure 4. 110: Cut lead defect of transistor	111
Figure 4. 111: Damaged case defect of transistor	111
Figure 4. 112: Misplaced defect of transistor	111
Figure 4. 113: Learning curve for zipper using DeeplabV3+ model	112
Figure 4. 114: Accuracy of the trained model for Zipper	113
Figure 4. 115: Predicted mask for the trained model for Zipper	114
Figure 4. 116: Fabric border defect of zipper	115
Figure 4. 117: Fabric interior defect of zipper	115
Figure 4. 118: Rough defect of zipper	115
Figure 4. 119: Squeezed teeth defect of zipper	116
Figure 4. 120: Split teeth defect of zipper	116

LIST OF TABLES

Table 3. 1: MVTec Anomaly Detection Datasets [49]	32
Table 3. 2: VGG16 Architecture Summary [52].....	34
Table 4. 1: Average Precision Recall for different objects	60
Table 4. 2: Accuracy for different objects using Custom U-Net model	89
Table 4. 3: Accuracy for different objects using DeeplabV3+ model	116
Table 5. 1: Summary of findings of this thesis	119

LIST OF SYMBOLS AND ABBREVIATIONS

ML	Machine Learning
CNN	Convolutional Neural Network
U-Net	U-Net Model for image segmentation
DeepLabV3+	DeepLabV3+ segmentation model
ASPP	Atrous Spatial Pyramid Pooling
HWC	Height, Width, Channel (data layout format)
CHW	Channel, Height, Width (data layout format)
RGB	Red, Green, Blue (color channels for images)
ReLU	Rectified Linear Unit (activation function)
FC	Fully Connected (layer in neural networks)
SoftMax	SoftMax function (used in classification tasks)
VGG16	Pre-trained Convolutional Neural Network model with 16 layers
ET	Epochs (used in training iterations)
AP	Average Precision
mIoU	Mean Intersection over Union (evaluation metric for segmentation)

Chapter 01

Introduction

1.1 Background and motivation

In manufacturing applications, the ability to detect faults is essential. Defect identification is necessary to make sure a manufacturing process is under control and operating as intended. Process performance can be maintained through the implementation of suitable corrective actions, contingent on the type and degree of detected anomalies. Another essential step in the inspection process before accepting or rejecting a part made by a process or supplied by a supplier is defect identification. Additionally, it can facilitate part rework and repair, hence minimizing material waste. If flaws are found early enough, a feedback control mechanism used in some manufacturing processes can be utilized to stop them from forming. In order to create process models that may be applied to process improvement, defect detection is also essential. Defect identification used to be done by human professionals who were familiar with the procedure. Automated flaw identification is necessary if manufacturing processes are to be more automated [1] .

In the context of quality control and defect detection, the terms "defect", "anomaly", and "fault" are often used interchangeably, but they have distinct meanings.

Defect: A defect refers to a non-conformance or an irregularity in a product or a process that does not meet the specified requirements or standards. Defects can be related to the material, design, manufacturing process, or functionality of a product. In the context of metallic products, defects can include cracks, scratches, corrosion, or other types of damage.

Anomaly: An anomaly refers to a data point or a pattern that deviates significantly from the expected or normal behavior. Anomalies can be indicative of defects, but not all anomalies are defects. Anomalies can be caused by various factors, such as measurement errors, noise, or unusual operating conditions.

Fault: A fault refers to a condition or a situation that causes a product or a process to fail or malfunction. Faults can be related to design, manufacturing, or operational

issues. In the context of metallic products, faults can include structural weaknesses, material defects, or other types of failures. While the terms "defect", "anomaly", and "fault" are related, they have distinct meanings:

- A defect is a non-conformance or irregularity in a product or process.
- An anomaly is a data point or pattern that deviates significantly from the expected behavior.
- A fault is a condition or situation that causes a product or process to fail or malfunction.

There are various points in the production process where product problems arise. This study argues that a smart factory can be constructed by integrating it into the small-scale, scarce product manufacturing process. This primarily addresses the issue of visual inspection for defect detection. The use of an intelligent manufacturing process can reduce human costs and minimize faults, hence facilitating sustainable growth [2].

Defect detection in industry is increasingly being entrusted to machine learning (ML) rather than human inspection due to its superior efficiency, accuracy, and scalability. Unlike human inspectors, who can experience fatigue, inconsistency, and are limited by their subjective judgment, ML algorithms can analyze vast amounts of data quickly and with high precision. These algorithms can identify subtle defects that might be missed by the human eye, ensuring higher quality control. Furthermore, ML models can continuously improve over time as they are exposed to more data, leading to more accurate predictions. This not only reduces the cost associated with defective products but also enhances overall production efficiency by allowing for real-time monitoring and swift corrective actions. Consequently, ML-driven defect detection is a vital component in modern manufacturing, ensuring that high standards are consistently met while minimizing human error.

Industry 4.0 includes smart manufacturing as a fundamental component. It has the ability to incorporate highly intelligent systems into production procedures. Time is saved, quality is raised, and needless waste is removed. We can use AI and ML to optimize and streamline these procedures, which will enable us to respond to changes

more quickly and effectively. Changes in product demand, customer requirements, or outside events might cause these adjustments at any point in the value chain [3].

These traditional machine learning techniques can only handle a certain amount of unstructured input in its raw form (such photos, natural language, etc.). It is also possible to automatically process and extract features from such raw data using contemporary, cutting-edge machine learning (ML) techniques that are based on deep learning (DL) [4], [5]. Deep learning algorithms can handle unstructured data, like color photographs, and are particularly good at finding complicated structures in massive volumes of data [6].

This study has been worked on MVTecAD Anomaly detection data sets. There are 15 categories in the MVTecAD Anomaly Detection dataset, with 1725 photos used for testing and 3629 images used for training and validation. There are only perfect photographs in the training set. Both photos with and without various kinds of flaws are included in the test set.

An overview of each object category is provided in the Figure below displays a few sample photos for each category along with an example of a defect. Ten categories depict different kinds of items, while five cover different kinds of regular (carpet, grid) or random (leather, tile, wood) textures. Certain objects, like bottles and metal nuts, have a fixed appearance and are rigid, but other objects, like cables, are flexible or have natural differences, like hazelnuts. Some objects, like the toothbrush, pill, and capsule, were obtained in a generally aligned stance, while other objects, like the metal nut, screw, and hazelnut, were randomly rotated in front of the camera. Test photos of anomalous samples show a range of flaws, including surface defects (dents, scratches), structural defects (distorted object pieces), and faults that show up as the absence of specific object sections. There are 73 distinct fault kinds in all, with five on average in each category. But this study focused on only metallic texture (Grid) and other objects (Cable, Metal nut, Screw, Transistor, Zipper) shown in (Figure1.1-Figure1.6) with 33 kinds of defect for 6 items [7][8].



Figure 1. 1: Cable [9]

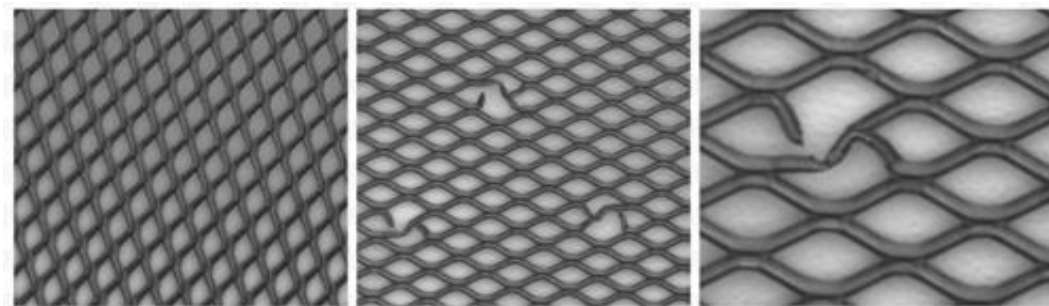


Figure 1. 2: Grid [9]

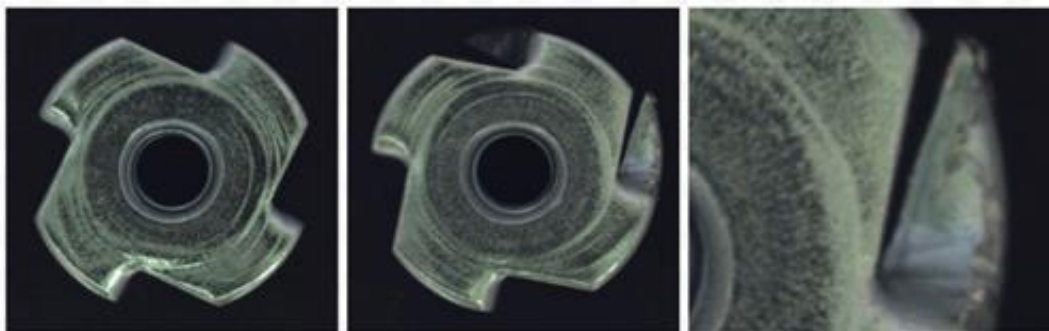


Figure 1. 3: Metal nut [9]



Figure 1. 4: Screw [9]

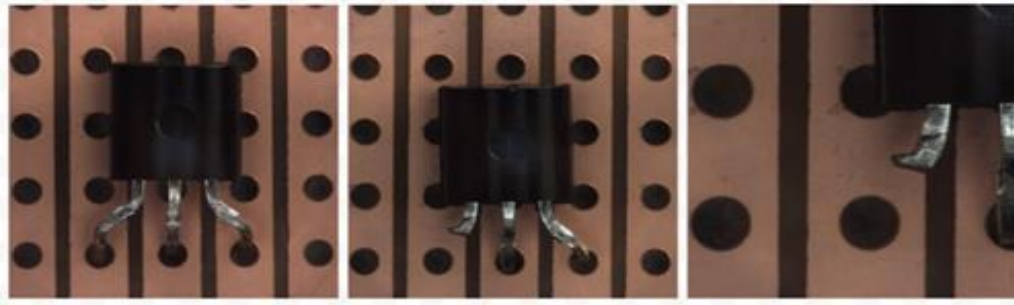


Figure 1. 5: Transistor [9]

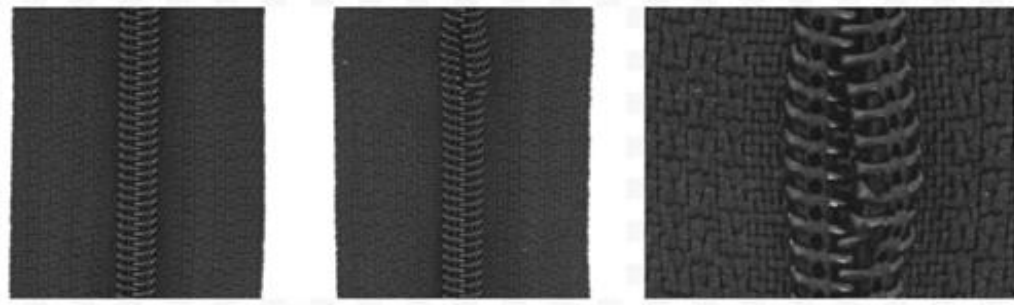


Figure 1. 6: Zipper [9]

1.2 Thesis objectives

1. To develop a machine learning model for detecting fault in metallic products.
2. To design and compare deep learning models for defect segmentation.
3. To propose industrial applications and future research directions for real-time Defect detection.

1.3 Context of defect detection on metallic product

In today's highly competitive global market, defects in manufacturing cause additional costs and these costs decrease the profit margin. So, defect-free products are crucial for firms. There are various studies conducted with different methodologies in different fields related to defect detection. This research aims to utilize machine learning algorithms for predicting defects in the manufacturing. Different ML algorithms are used to predict defected products [10].

Data sets for MVTecAD anomaly detection have been used in this study. The dataset for MVTecAD Anomaly Detection is divided into 15 categories. However, the mesh and other objects (cable, metal nut, screw, transistor, zipper) depicted in the image

below with 33 different types of defects for six items were the only subjects of this investigation [7].

This study's investigation of the use of Mini Batch Dictionary Learning with Sparse Coder is one of its contributing elements. The images are processed by Mini Batch Dictionary Learning with Sparse Coder algorithm and that can detect defect of the above 6 six metallic products.

1.4 Context of segmentation of metallic product surface defects using Custom U-Net model

Defect detection on metal surfaces is crucial for quality control and precise maintenance in various industrial scenarios. Nevertheless, metal surface defects demonstrate various variations and intricacies, making identifying high-precision metal surface defects a formidable task. With the development of artificial intelligence technology, metal surface defect detection technology has ushered in a huge breakthrough. Compared to manual inspection, metal surface defect detection methods based on computer vision and artificial intelligence technology have apparent advantages in detection efficiency and accuracy, becoming the mainstream means of detecting surface defects in metal parts [11].

Recently a substantial advancement has been noticed in deep learning, which has led to several studies using deep learning techniques in defect identification. Metal surface fault identification has become much more accurate and efficient [12], [13], [14].

One of the contributing factors of this research is its examination of the application of deep learning-based customized U-Net model to segment the surface defect of metallic products.

1.5 Context of segmentation of metallic product surface defects using DeepLabv3+ model

The DeepLabV3+ model is widely recognized for its effectiveness in semantic segmentation tasks, especially for complex surface textures like metallic products. Its key feature is the Atrous Spatial Pyramid Pooling (ASPP), which helps capture features at multiple scales, making it highly effective for identifying subtle surface

defects such as cracks, scratches, and structural deformities. The model operates with an encoder-decoder architecture, where the encoder captures rich semantic information and the decoder refines the segmentation for more precise results. This is particularly useful in industrial applications where defect detection and quality control are critical [15].

In various studies, DeepLabV3+ has been employed to segment surface defects in metallic products, including cables, grids, and screws, demonstrating excellent performance in pixel-wise segmentation. For example, when applied to steel surface defect detection, the model—especially when combined with backbones like ResNet101—achieved a high mean Intersection over Union (mIoU), significantly outperforming traditional methods in terms of both accuracy and processing speed. This capability ensures that even minor imperfections can be detected efficiently, which is crucial for maintaining high standards in manufacturing processes.

The versatility of DeepLabV3+ has been further enhanced through the use of data augmentation techniques, which address the challenges of limited and imbalanced datasets. For instance, studies on the NEU-seg dataset for metallic surface defects have shown that augmenting the dataset with synthetic images generated by diffusion models further improved the generalization of the DeepLabV3+ model, leading to more robust and accurate segmentation results [16].

By incorporating DeepLabV3+ into production workflows, manufacturers can significantly reduce reliance on manual inspection, thereby decreasing human error and improving overall efficiency in defect detection and product quality control.

1.6 Conclusion

In conclusion, the defect detection and segmentation of defect in metallic products are vital in industrial manufacturing for ensuring product quality, safety, and operational efficiency. Deep learning algorithms have revolutionized this process by providing automated, accurate, and real-time inspection solutions. These models can detect surface defects with greater precision than manual inspections, reducing human error

and labor costs. By enabling pixel-wise defect segmentation, deep learning helps in identifying even subtle anomalies that traditional methods might miss. This not only improves product quality but also optimizes production workflows by minimizing waste, preventing faulty products from advancing in the manufacturing cycle, and reducing downtime. As industries embrace automation and Industry 4.0, integrating deep learning for defect detection enhances scalability, productivity, and ensures compliance with safety standards, particularly in sectors like aerospace, automotive, and electronics, where quality is paramount.

The organization of this thesis is structured in a total of five chapters. The next chapter illustrates all the previous studies and literature regarding defect detection and segmentation in metallic products. Chapter 3 discusses the detailed methodology of how the work is done. Chapter 4 represents all the results obtained from implementing various machine learning algorithms. Also, the results of ML models are discussed in detail. The last chapter of this thesis throws light on the conclusion of the work and future implementations and research directions.

Chapter 02

Literature Review

2.1 Introduction

The initial phase of this research endeavor involves a comprehensive analysis of studies about the defect detection and segmentation of defect in metallic products using different types of ML models, such as Mini Batch Dictionary Learning, Customized U-Net. This preliminary investigation aims to offer a thorough understanding of the methodologies employed by previous researchers. Thus, the identified limitations and gaps in previous research will play a vital role in shaping the objectives fulfillment and methodologies of this thesis.

2.2 Previous study about defect detection and segmentation of defect in metallic products using various ML algorithms

This section presents all the previous literature regarding machine learning algorithms. Xiaoqing et al. [17] presented a review of surface defect inspection methods, emphasizing the role of deep learning in improving automated visual inspection (AVI) systems for industrial quality control. The paper covers prior literature on vision-based surface inspection, traditional algorithms, and recent deep learning advancements. It specifically focuses on applications in semiconductor, steel, and fabric manufacturing, while also providing information on publicly available datasets to support further research in this area. The review aims to highlight both traditional and state-of-the-art deep learning approaches to enhance defect detection.

Chung-Chi Huang et al. [18] proposed a machine learning-based intelligent defect detection system for metal products, addressing common surface and size defects. The system consists of three parts: (1) Image Preprocessing using OpenCV for preparing images, (2) Training Procedures with CNN and GAN to enhance defect detection accuracy, especially with small datasets, and (3) Testing Procedures implemented in

Python on GPU-based hardware. The system improves detection accuracy through GAN-based data augmentation and reduces the reliance on costly manual inspection. It contributes to advancing Industry 4.0, enhancing industrial productivity, and addressing labor shortages due to aging populations.

Adriana et al. [19] presented an automated defect detection system for porcelain products, developed in collaboration with an industrial producer. The system integrates robotics, computer vision, and machine learning, specifically using a convolutional neural network (CNN) to analyze product images and identify defects. This approach addresses the current reliance on manual inspection in the porcelain industry. Experimental results demonstrate promising performance, and the system is expected to improve production efficiency and reduce costs, delivering economic benefits to the company.

Jing Yang et al. [20] introduced a real-time defect detection method for tiny parts in intelligent production. It addresses challenges in manually adjusting conveyor speed and camera settings. The study analyzes the impact of part properties and environmental factors on detection stability and develops a correlation model between detection performance and conveyor speed. A deep learning-based Single Shot Detector (SSD) algorithm is proposed to detect defects, while an additional algorithm helps reduce missed detections. Experiments with a 0.8 cm darning needle showed the highest detection accuracy at a conveyor speed of 7.67 m/min.

Xian Tao et al. [21] proposed an advanced method for automatic metallic surface defect detection in industrial settings, addressing the limitations of traditional approaches. It introduces a two-step process: a cascaded autoencoder (CASAE) for segmenting and localizing defects, followed by a convolutional neural network (CNN) for classifying the defects. This method successfully detects metallic defects under various conditions, showing robustness and accuracy in experiments with industrial datasets. It can also be extended to other defect detection applications.

Mohammad et al. [22] developed a CNN-based deep learning model to detect real-time defects in 3D printing, a critical aspect of Industry 4.0. The model identifies issues such as inconsistent extrusion, weak infills, and sagging by comparing the geometrical features of printed parts to a perfect standard. Integrated with a camera on the 3D printer, the system processes images to detect anomalies, aiming to optimize and automate the 3D printing process, reduce production losses, and minimize human involvement in quality control.

Enes et al. [23] investigated the use of machine learning algorithms to predict manufacturing defects, aiming to improve product quality and reduce costs. The study evaluates various ML algorithms for their effectiveness in forecasting defects during the powder coating phase at a major Turkish manufacturing company. It compares the performance of these algorithms to identify the most suitable one for predicting defective products and enhancing quality control.

Praveen et al. [24] proposed a Transfer Learning-based U-Net (TLU-Net) framework for detecting steel surface defects, aiming to enhance automated visual inspection (AVI) methods. It compares U-Net models with ResNet and DenseNet encoders using both random initialization and pre-trained networks on ImageNet. Experiments with Severstal data reveal that transfer learning improves defect classification by 5% and defect segmentation by 26% over random initialization, with greater benefits as training data decreases.

Saksham et al. [25] introduced a novel data augmentation framework using Generative Adversarial Networks (GANs) to create synthetic images for surface defect detection in manufacturing. This approach addresses the challenge of acquiring large annotated datasets. GAN-generated images improve the training of classification algorithms, leading to enhanced performance. Tested on the NEU-CLS dataset for steel strip defects, the GAN-augmented CNN achieved better sensitivity (95.33%) and specificity (99.16%) compared to traditional methods. The framework demonstrates strong generalization potential and can be applied to other surface inspection tasks, advancing automated defect detection in manufacturing.

Weidong et al. [26] presented an improved target detection algorithm for identifying small and complex steel surface defects. By enhancing the traditional Faster R-CNN algorithm with multiscale fusion and a deformable convolution network, the authors address challenges in detecting small targets. Tested on the NEU-DET dataset, the algorithm achieves a mean average precision (mAP) of 0.752, a 0.128 improvement over the original model. This method effectively detects small steel surface defects, offering a useful solution for automated steel defect detection systems.

Domen et al. [27] introduced a deep-learning architecture for detecting and segmenting surface anomalies, specifically cracks, with minimal training data. It is designed to be practical for industrial applications, requiring only 25-30 defective samples for training, unlike typical deep-learning models that need hundreds or thousands of samples. The proposed model outperforms existing methods, including commercial software, in surface-crack detection. The study also examines annotation precision, required training samples, and computational costs. Additionally, the dataset used is publicly available to support further research in surface-defect detection.

Swarit et al. [28] developed a machine vision-based framework for detecting surface defects in manufacturing using a pre-trained Convolutional Neural Network (ResNet-101) combined with a Support Vector Machine (SVM) classifier. Aimed at minimizing the need for large datasets and computational resources, the study focuses on surface defect detection during the centerless grinding of tapered rollers. Using image data augmented for better training, the system achieved 100% precision for defect-free components. The framework is designed to be efficient for Micro, Small, and Medium Enterprises (MSMEs) and Small and Medium-sized Enterprises (SMEs), providing an automated alternative to manual inspection techniques.

Demir [29] et al. presents a deep learning-based method for detecting and classifying surface defects in steel production. The approach consists of four steps: (1) a Parallel Attention Residual-Convolutional Neural Network (PAR-CNN) is designed to enhance classification performance, (2) deep features are extracted from the trained model, (3) a new feature selection algorithm (NCA-Relief Matched Index) is introduced to select the most relevant features, and (4) these selected features are classified using a Support Vector Machine (SVM). The method was tested for both binary and multi-class classification on the Severstal: Steel Defect Detection dataset from Kaggle.

Huang et al. [30] introduced an efficient convolutional neural network (CNN) for fabric defect detection and segmentation in the textile industry, addressing key challenges such as detection accuracy, computational complexity, and data imbalance. The proposed framework reduces the need for extensive manual annotation, requiring only around 50 defect samples along with standard samples to learn defect features and accurately locate them. The network consists of two parts: segmentation and decision. First, an untrained fabric dataset is fed into the segmentation network, whose output is then used to train the decision network. The well-trained network is capable of real-time defect detection at 25 FPS. Experimental results on a public and three custom fabric datasets show that the method outperforms eight state-of-the-art approaches in accuracy and robustness

Liong et al. [31] proposed a fully automated defect detection and marking system for calf leather using a deep learning-based machine vision method. The system, consisting of a high-resolution camera, LED light, and a robot arm, identifies leather defects without human intervention (except during the defect ground truth construction). The process involves capturing leather images, training and testing these images using a deep learning architecture, and using mathematical geometry to determine defect boundaries. The algorithm achieves a segmentation accuracy of 91.5% on training data and 70.35% on test data. Additionally, performance metrics like the confusion matrix, F1-score, precision, specificity, and sensitivity are reported to confirm its effectiveness.

Yunpeng Wu et al. [32] introduced a novel Rail Boundary Guidance Network (RBGNet) for detecting rail surface defects (RSDs) with high precision. Current detection methods face challenges such as low accuracy and blurry rail edges (RE). To address these, RBGNet fully leverages the complementarity between rail surface (RS) and rail edge (RE) features. It progressively fuses high-level RS information with shallow RE edge features to enhance edge clarity. The model integrates refined edge features with RS data at various levels to improve detection accuracy. Additionally, a hybrid loss function combining binary cross entropy (BCE), structural similarity index measure (SSIM), and intersection-over-union (IoU) is used to supervise the learning process and improve RS location and edge refinement. The system also includes an image-based model for automated RSD detection and quantification. Experiments using a complex UAV rail dataset demonstrate that the proposed method achieves a high detection rate with robust performance in challenging environments.

Du-Ming et al. [33] developed a two-stage deep learning approach for automatic defect detection in material surfaces, addressing the challenge of pixel-wise annotation for training semantic models. The first stage utilizes two CycleGAN models to automatically synthesize defect images and generate pixel-wise annotations, eliminating the need for manual annotation. These synthesized images and their corresponding annotations are then used to train a U-Net semantic network for defect segmentation. The scheme requires only a small set of real defect samples for training and is highly practical and computationally efficient for industrial use. Experimental results demonstrate that this method achieves high detection accuracy across various textured and patterned surfaces.

Okaro et al. [34] focused on the automatic detection of faults in additive manufacturing (AM), particularly in laser powder-bed fusion (L-PBF), using a semi-supervised machine learning approach. The method involves capturing real-time process images and applying machine learning algorithms to detect anomalies in AM parts. Their model combines labeled and unlabeled data to improve Defect detection without requiring a fully annotated dataset. While this reduces the need for extensive data

labeling, the method still struggles with handling highly complex geometries and varying surface textures, particularly when the training dataset lacks diversity. Their results indicated a significant improvement in defect detection accuracy, particularly in reducing false negatives. However, limitations included difficulties in generalizing across different AM processes and a dependence on image quality for accurate fault identification.

Kang et al. [35] conducted a systematic literature review that explored various machine learning applications in production lines, particularly focusing on fault diagnosis and quality control. They evaluated the performance of several machine learning algorithms, such as decision trees, support vector machines (SVM), and neural networks, across different production settings. One of the key insights from their review was that while machine learning techniques have substantially improved Defect detection, their real-time application remains limited due to computational complexity and slow adaptability to changing production conditions. Many algorithms also require large amounts of high-quality data, which is not always feasible in dynamic production environments. The review highlighted the need for more flexible machine learning models that can handle real-time data, adapt to changing variables, and work with smaller datasets, especially in small-scale manufacturing setups.

Westphal and Seitz et al. [36] introduced a machine learning method specifically designed for defect detection and visualization in selective laser sintering (SLS), an additive manufacturing process. They implemented deep learning models, particularly convolutional neural networks (CNNs), to analyze process images captured during SLS. The CNNs were trained on these images to detect defects such as cracks and surface imperfections. Their methodology involved preprocessing the data to filter noise and enhance feature extraction. The study reported a significant improvement in detecting defects compared to traditional methods, with a 10% increase in detection accuracy. However, one major limitation is the difficulty in generalizing the model across different types of metals and materials used in SLS, as the trained CNNs are highly specific to the data they were trained on. Additionally, their model struggled

with smaller or more intricate defects, which were often missed unless the input image quality was exceptionally high.

Tao et al. [37] developed an automated system for detecting metallic surface defects using convolutional neural networks (CNNs). Their research compared the performance of CNNs with traditional machine learning algorithms that rely on manually engineered features. The CNN-based system significantly outperformed traditional methods, achieving a 15% higher detection accuracy. The authors emphasized the advantages of CNNs in their ability to automatically learn features directly from the data, bypassing the need for manual feature extraction. However, the system had limitations in detecting smaller defects, particularly in cases where defects were subtle or blended into the surface textures. The success of the CNN-based system was highly dependent on the quality and resolution of the input images, and the authors noted that low-quality images led to a substantial drop in performance. Furthermore, the model required a significant amount of labeled data for training, which may not be readily available in many industrial settings.

Park et al. [38] provided a comprehensive review of Defect detection and process diagnostics in industrial settings, comparing model-based and machine learning-based approaches. The review covered traditional statistical methods and newer machine learning techniques, such as neural networks and random forests. Machine learning methods, while faster and more scalable, were found to lack robustness in environments with high variability, such as semiconductor manufacturing. The authors argued that while machine learning models excel at identifying known defects, they struggle with detecting previously unseen faults. A limitation of their review was the relatively small number of studies addressing the issue of adapting machine learning models to rapidly changing environments. They suggested that future research should focus on developing models capable of real-time learning and adaptation, as current methods are often constrained by the need to retrain models frequently as conditions change.

Yun et al. [39] proposed a deep learning-based system for detecting defects on metal surfaces. Their system utilized data augmentation techniques to generate additional training samples from a limited dataset. By incorporating this data augmentation process, they were able to train their deep learning model more effectively, compensating for the small size of their original dataset. Their methodology involved applying a CNN architecture to identify defects, and the results showed a detection accuracy of 94%. This high accuracy is particularly notable given the challenges of detecting defects in noisy and inconsistent data. However, the system's limitations included a lack of validation in real-time industrial applications. While the method performed well in controlled experiments, its performance in real-world conditions, where production speeds and environmental factors vary, remains uncertain. Additionally, the computational cost of the system is high, making it less suitable for resource-constrained environments.

Yang et al. [40] conducted a comprehensive survey on the use of deep learning for detecting defects in manufacturing processes. They reviewed state-of-the-art deep learning models, including CNNs and generative adversarial networks (GANs), focusing on their application to defect detection in production lines. Their analysis found that deep learning models offer superior pattern recognition capabilities compared to traditional machine learning methods. However, the survey also highlighted the limitations of these models, particularly their reliance on large annotated datasets, which can be difficult and expensive to obtain. Furthermore, the computational demands of training deep learning models make them impractical for smaller manufacturing facilities with limited processing power. Another limitation is the lack of real-time adaptability in most deep learning approaches, which struggle to update their models in response to new data without retraining from scratch.

Rai et al. [41] investigated the role of machine learning in Industry 4.0, focusing on predictive maintenance and fault diagnosis. Their research explored various machine learning algorithms, such as decision trees, SVM, and random forests, for predicting equipment failures and ensuring product quality in manufacturing environments. The study emphasized the growing importance of predictive maintenance in minimizing

downtime and reducing operational costs. However, a key challenge they identified is the scalability of machine learning models when applied to large, heterogeneous datasets, which are common in industrial settings. The authors noted that while machine learning offers significant benefits in predictive accuracy, the high computational costs and the need for extensive training data remain significant barriers to widespread adoption in small- to medium-sized enterprises.

Yang et al. [42] developed a real-time system for detecting defects in tiny mechanical parts, focusing on parts such as screws and darning needles. Their method combined image processing techniques with deep learning, specifically using CNNs to detect surface anomalies. The system was designed to function in real-time, with the goal of minimizing delays in production lines. Their experiments showed high detection accuracy, particularly at conveyor speeds of 7.67 m/min. However, the system struggled under challenging environmental conditions, such as low lighting and vibrations, which are common in industrial settings. The authors highlighted the need for more robust models that can maintain high accuracy in varying conditions without sacrificing speed or computational efficiency.

Khan et al. [43] used machine learning for real-time defect detection in 3D printing, specifically in selective laser melting (SLM). Their approach involved integrating thermal imaging with a deep belief network (DBN) to detect flaws in printed parts. The system monitored thermal profiles during the printing process, comparing the actual profile to an idealized standard to identify defects such as cracks or incomplete layers. The results showed that the DBN achieved a detection rate of over 90%, making it a promising solution for real-time quality control in additive manufacturing. However, the system faced challenges with more complex geometries and dynamic thermal profiles, where slight variations could result in false positives. Additionally, the system's reliance on thermal imaging requires precise calibration, which may not be practical for all manufacturing environments.

DeepLabV3+, an advanced deep learning architecture designed for semantic image segmentation, has shown great promise in various fields, including defect detection in industrial processes. Its ability to capture and label fine details in images makes it particularly useful for identifying defects on metallic surfaces, where precision is crucial. Several studies have demonstrated the potential of DeepLabV3+ for Defect detection in metallic products.

Zhu R et al. [44] demonstrated the effectiveness of DeepLabV3+ in fabric defect detection, offering a robust alternative to traditional machine vision algorithms. The performance parity across ResNet-18, ResNet-50, and MobileNetV2 suggests that lightweight models like MobileNetV2 can perform on par with deeper networks, which is beneficial for real-time applications where computational efficiency is crucial. This study contributes to the growing body of work on deep learning-based defect detection, offering new methodologies and technical insights that pave the way for more automated, accurate, and efficient fabric inspection systems.

Nie Z et al. [16] highlighted the importance of backbone selection in DeepLabV3+ models for defect detection in steel images. While all backbones demonstrated the ability to segment defects, ResNet101 emerged as the best option due to its superior IoU score and faster training time, making it ideal for real-time industrial applications. EfficientNet also showed competitive performance, while DenseNet lagged behind in both segmentation accuracy and IoU scores. These insights contribute to the growing body of research on improving defect detection in steel manufacturing using deep learning, with practical implications for enhancing production quality and efficiency.

Emre Güçlü et al. [45] modified DeepLabV3+ architecture has been proposed to improve the segmentation performance of steel surface defects. The study introduces the Squeeze-and-Excitation (SE) module and enhances the Atrous Spatial Pyramid Pooling (ASPP) module within the DeepLabV3+ model, resulting in significant improvements in the mean Intersection over Union (mIoU) metric. Experimental results show that these modifications lead to better segmentation performance compared to other popular models. This advancement demonstrates the potential of

deep learning in enhancing defect detection accuracy and reliability in the steel industry.

Qianqian Zhou et al. [46] introduced a novel DeepLabV3+-based approach for pixel-level segmentation and severity quantification of sewer defects, comparing its performance with SegNet, FCN, and U-Net models. The research highlights that DeepLabV3+ with a ResNet-50 backbone outperforms the other methods in accurately identifying various sewer defects, achieving high precision with PA, mIoU, fwIoU, and F1-score values of 0.9, 0.53, 0.84, and 0.55, respectively. While the model was most effective in detecting residential wall defects, followed by disjoints and tree roots, it showed some limitations in accurately quantifying severity, with 30% of predictions overestimating defect severity. Overall, the method offers a robust solution for improving the accuracy and efficiency of sewer system inspections.

Yun Zuh et al. [47] proposed a multi-class segmentation model based on an improved DeepLabv3+ to overcome limitations in the accuracy and precision of existing semantic segmentation methods. Traditional approaches struggle with rough edge segmentation, poor performance on small defect detection, and loss of deep-level semantic information, leading to suboptimal feature extraction. The proposed model incorporates a Coordinate Attention Mechanism into the DeepLabv3+ framework to enhance feature retention and improve segmentation performance, particularly for small and complex defects. This advancement aims to boost the accuracy and reliability of automated fruit quality inspection systems.

Ruixin Dong et al. [48] proposed an improved DeepLabv3+ algorithm combined with transfer learning. The original DeepLabv3+ backbone (Xception) is replaced with a lightweight MobileNetv2 network to reduce model parameters and enhance inference speed. The model further integrates heterogeneous sensory field fusion and void depth separable convolution to optimize the atrous spatial pyramid pooling structure, improving information utilization and training efficiency. Additionally, an attention mechanism and data augmentation strategies are employed to boost network training performance and defect recognition accuracy. Experimental results demonstrate that the improved model significantly reduces iteration steps, increases training speed, and

improves the mean Intersection over Union (mIoU) by 3.46%. The number of model parameters is reduced by 69.8%, and the model shows a marked improvement in defect edge segmentation, making it effective for fast, real-time roller surface defect detection with notable practical significance.

Dmitry A. Yudin et al. [49] investigates the use of deep neural networks for segmenting flat roof defects in aerial images, focusing on five defect categories: "hollows," "swelling," "folds," "patches," and "breaks." Three architectures—U-Net, DeepLabV3+, and HRNet+OCR—are evaluated using the newly introduced RoofD dataset, which contains 6,400 image pairs of aerial photos and corresponding ground truth masks. Various training approaches are analyzed, including the SDice coefficient with categorical cross-entropy for U-Net and the lightweight U-NetMCT architecture, as well as weighted categorical cross-entropy for DeepLabV3+ and HRNet+OCR. The multi-scale HRNet+OCR model achieves the highest mean Intersection over Union (mIoU) score of 0.44, while U-NetMCT and DeepLabV3+ offer faster inference times but lower mIoU scores (0.33–0.37). "Patches" are identified as the most challenging category due to their rarity in the dataset. The paper also highlights the potential application of these models in software designed to automate roof defect detection for industrial and municipal uses.

Upon reviewing the existing literatures and the current study, several research gaps have been identified that warrant further investigation.

1. Limited exploration of metallic product defect detection: While there have been studies on defect detection in various industries, the specific application of metallic product defect detection has received limited attention. This study has made a significant contribution to this area, but further research is needed to explore the nuances of metallic product defect detection.

2. Insufficient investigation of anomaly detection techniques: The current study employed Mini Batch Dictionary Learning with Sparse Coder for anomaly detection, but other techniques, such as One-Class SVM, Local Outlier Factor (LOF), and

Isolation Forest, have not been thoroughly explored in the context of metallic product defect detection. A comprehensive comparison of these techniques is necessary to determine their effectiveness.

3. Lack of research on real-world datasets: The current study utilized the MVTec AD dataset, which is a benchmark dataset for anomaly detection. However, real-world datasets from industrial settings are scarce, and there is a need for more research on defect detection using real-world data.

4. Inadequate exploration of deep learning architectures: While the current study employed Custom U-Net and Deeplab V3+ for defect segmentation, other deep learning architectures, such as ResNet, Inception, and DenseNet, have not been thoroughly explored in the context of metallic product defect detection. A comprehensive comparison of these architectures is necessary to determine their effectiveness.

5. Limited investigation of transfer learning: The current study employed transfer learning for Deeplab V3+, but the effectiveness of transfer learning for other deep learning architectures and defect detection tasks has not been thoroughly explored. Further research is needed to investigate the potential benefits and limitations of transfer learning in metallic product defect detection.

6. Insufficient research on defect detection in complex scenes: The current study focused on defect detection in relatively simple scenes, but real-world industrial settings often involve complex scenes with multiple objects and classes. Further research is needed to develop effective defect detection methods for complex scenes.

7. Lack of investigation on the impact of data quality: The current study assumed high-quality data, but in real-world industrial settings, data quality can vary significantly. Further research is needed to investigate the impact of data quality on defect detection performance and to develop methods that can handle noisy or low-quality data.

Future Research Directions

Based on the identified research gaps, future studies should focus on:

1. Exploring other anomaly detection techniques and comparing their effectiveness.
2. Developing and evaluating deep learning architectures for defect detection in metallic products.
3. Investigating transfer learning for various deep learning architectures and defect detection tasks.
4. Developing methods for defect detection in complex scenes.
5. Investigating the impact of data quality on defect detection performance and developing methods to handle noisy or low-quality data.

By addressing these research gaps, future studies can contribute to the development of more effective and robust defect detection methods for metallic products, ultimately improving the efficiency and quality of industrial manufacturing processes.

2.3 Conclusion

The literature reviewed in this chapter highlights the significant advancements and challenges in defect detection and segmentation for metallic products using machine learning algorithms. Previous studies have demonstrated the transformative potential of machine learning, particularly deep learning approaches, in automating industrial defect detection, improving accuracy, and reducing human errors. Techniques such as CNN, U-Net, and GAN have shown marked improvements in identifying defects with precision, especially in metal manufacturing. However, the literature also underscores some limitations, particularly the reliance on large datasets, computational resources, and real-time adaptability in industrial applications. While the accuracy of these machine learning models is promising, achieving scalability and robustness in diverse and changing environments remains a critical challenge. This review has established a solid foundation for the methodologies adopted in this thesis, aimed at enhancing defect detection processes by addressing the gaps identified in prior research, particularly regarding the application of Mini Batch Dictionary Learning and Custom U-Net models.

Chapter 03

Methodology

3.1 Introduction

The first part of the methodology presents a machine-learning approach mainly Mini Batch Dictionary Learning with Sparse Coder for defect detection. For any machine learning model to work, one needs to follow certain steps for successful completion of the work. All the steps begin with the data collection process. In the later part, with the aim to segmentation of defect on metallic products using Custom U-Net model and DeeplabV3+ model.

3.1.1 Image Processing Algorithms Employed

The research utilizes a range of image processing algorithms designed to prepare, analyze, and enhance image data for effective defect detection and segmentation. These algorithms play a critical role in ensuring that the input data is optimally processed to improve model performance and segmentation accuracy. The main image processing algorithms employed in this study are outlined below:

1. Image Preprocessing and Normalization: Preprocessing involves standardizing the images to a consistent format and size, which is crucial for deep learning models. Each image is resized to a specific resolution to maintain uniformity, ensuring that variations in image dimensions do not impact model performance. Additionally, grayscale images are converted to RGB to expand the color channels, enabling more detailed analysis by models like U-Net and DeepLabV3+. Pixel values are normalized to scale intensity values between 0 and 1, reducing the computational load and enhancing the model's convergence during training.

2. Data Augmentation Techniques: Data augmentation is employed to artificially expand the dataset and introduce variability, which helps in reducing overfitting and

improving model generalization. Techniques such as random rotations, flips, and zooms are applied to create variations in the image dataset, simulating different defect orientations and perspectives. This augmentation strategy is particularly beneficial in defect detection, as it allows the model to become robust against minor distortions in defect appearance.

3. Feature Extraction Using VGG16: VGG16, a convolutional neural network pre-trained on ImageNet, is integrated into the system as a feature extractor. This transfer learning approach enables the model to leverage pre-learned hierarchical features that capture essential characteristics of metallic textures. The VGG16 layers extract low-level features such as edges and textures and higher-level representations that are useful for identifying patterns related to metallic defects. This pre-trained feature extraction significantly reduces the need for extensive labeled data, allowing for efficient model training and enhanced performance.

4. Sparse Representation through Mini Batch Dictionary Learning: Mini Batch Dictionary Learning is applied to represent images in terms of sparse bases (atoms), creating a compact representation of normal patterns in metallic products. This sparse coding helps isolate anomalies in the image data, highlighting areas that deviate from the expected structure. Sparse representations reduce computational complexity by focusing on essential features, facilitating efficient anomaly detection on large datasets without the need for extensive computational resources.

5. Multiscale Feature Extraction with Atrous Convolutions: The DeepLabV3+ model incorporates Atrous convolutions, also known as dilated convolutions, which capture features at multiple scales. This technique is crucial for identifying small and large defects within metallic textures, as it allows the model to gather information across various spatial scales without reducing the image resolution. By preserving detailed spatial information, Atrous convolutions improve the model's ability to localize defects precisely.

6. Binary Mask Generation for Defect Segmentation: The U-Net and DeepLabV3+ models generate binary masks that distinctly segment defective regions from non-defective areas. The binary mask highlights defect boundaries, allowing for precise localization of anomalies. This segmentation approach is fundamental to accurate defect detection, as it enables pixel-level analysis that distinguishes even subtle imperfections on metallic surfaces. The binary masks provide clear defect outlines that aid in further analysis and classification.

These image processing algorithms collectively support a structured workflow for defect detection and segmentation, ensuring high data quality and enabling the machine learning models to achieve accurate, reliable outcomes. Through preprocessing, feature extraction, sparse representation, and multiscale analysis, the system achieves a level of precision necessary for effective quality control in industrial applications.

3.1.2 The reasons behind choosing the ML algorithms are such as:

1. Mini Batch Dictionary Learning

- **Sparse Representation for Efficient Detection:** Mini Batch Dictionary Learning is selected for its capability to create sparse representations that effectively isolate anomalies within metallic images, making it suitable for detecting rare or subtle defects. By working on small data subsets, this approach enables efficient handling of large datasets without requiring extensive memory resources.
- **Scalability and Flexibility:** Operating on mini-batches allows for faster updates and easier scaling, essential for real-time or large-scale applications in industrial environments. This method helps in achieving high detection rates across varied metallic objects like grids, cables, and screws.
- **Compatibility with Sparse Coder:** Using a pre-trained dictionary, the Sparse Coder generates efficient representations for incoming data, ensuring minimal computational overhead while retaining significant image features. This

process supports unsupervised anomaly detection, suitable for applications where labeled defect data may be limited.

2. Custom U-Net

- **High-Resolution Segmentation:** Custom U-Net is optimized for defect segmentation, specifically at the pixel level, which is crucial for accurately identifying defect boundaries on metallic surfaces. Its encoder-decoder structure is effective for capturing spatial hierarchies within images, making it ideal for segmenting complex shapes and textures in metallic products.
- **Flexibility for Varied Objects:** By training a separate model for each metallic object type, the Custom U-Net adapts to distinct structural and surface characteristics (like those of grids, zippers, and transistors). This adaptability enhances segmentation accuracy and allows for detailed analysis of defects across different products
- **Generalization on Limited Labeled Data:** U-Net performs well with limited labeled data, achieving high accuracy (up to 99% for some objects) while maintaining consistency across unseen samples. This is beneficial in manufacturing settings where extensive labeled defect data may not be available for every defect type.

3. DeepLabV3+

1. **Advanced Feature Extraction for Complex Surfaces:** DeepLabV3+ is specifically chosen for its Atrous Spatial Pyramid Pooling (ASPP) module, which captures multiscale features within images. This is particularly useful in distinguishing fine details in complex metallic textures, improving segmentation precision for intricate defects.
2. **Enhanced Boundary Detection:** The encoder-decoder architecture in DeepLabV3+ refines coarse segmentation outputs, enabling it to handle precise edge detection around defects, which is essential for accurately delineating defect regions on metallic surfaces.
3. **Adaptability for Real-Time Applications:** DeepLabV3+ demonstrates the ability to segment defects in real-time applications, an advantage for automated quality control processes. This model offers a balance of high segmentation

accuracy and computational efficiency, particularly valuable in manufacturing settings where immediate feedback is critical.

Each of these algorithms contributes uniquely to achieving precise, efficient, and scalable defect detection and segmentation, essential for maintaining high-quality standards in metallic product manufacturing.

3.1.3 Overview of the system

1. System Architecture

The architecture of the defect detection system is designed to integrate machine learning and deep learning models for precise defect identification and segmentation in metallic products. This architecture comprises several critical components, including data input and preprocessing, feature extraction, anomaly detection, and pixel-level segmentation for comprehensive defect analysis.

- **Data Input and Preprocessing:** The initial phase of the system involves loading image datasets from the MVTec Anomaly Detection (AD) dataset, which includes a variety of metallic objects such as cables, grids, metal nuts, screws, transistors, and zippers. These images undergo a preprocessing pipeline to ensure compatibility with deep learning models. Preprocessing tasks include resizing images to standard dimensions, converting grayscale images to RGB, normalizing pixel values, and transforming the images into tensor format compatible with PyTorch. Additionally, VGG16, a pretrained convolutional neural network on ImageNet, is utilized as a feature extractor to capture complex patterns and textures specific to metallic surfaces, enhancing the model's ability to identify subtle yet significant details indicative of defects.
- **Model Integration and Execution:** The system adopts a multi-model approach, comprising Mini Batch Dictionary Learning for anomaly detection and custom U-Net and DeepLabV3+ models for pixel-level defect segmentation. Each model has been fine-tuned for a specific role within the system: Mini Batch Dictionary Learning is responsible for identifying anomalies in metallic textures, while U-Net and DeepLabV3+ facilitate precise segmentation of defect regions at a granular level. These models operate

collaboratively, with each output feeding into subsequent stages of the defect analysis process, thereby ensuring a comprehensive approach to identifying and categorizing defects in metallic products.

This layered system architecture leverages both unsupervised anomaly detection and supervised segmentation techniques, ensuring adaptability, precision, and efficiency in industrial settings where defect detection is paramount.

2. Configuration of the System

The configuration of the defect detection system has been carefully optimized to manage the high computational requirements associated with processing large volumes of high-resolution images and deploying advanced deep learning models. The configuration includes hardware and software components, each selected to ensure optimal performance, accuracy, and scalability in defect detection tasks for metallic products.

Hardware Configuration

To address the computational intensity of deep learning algorithms, the system operates in a GPU-accelerated environment, leveraging hardware such as NVIDIA GPUs. This configuration facilitates parallel processing, which is crucial for handling extensive computations involved in both training and deploying models like U-Net and DeepLabV3+ for image segmentation. The use of GPU-based infrastructure significantly reduces processing times, enabling real-time defect detection capabilities. Such a setup is essential in industrial applications where timely and efficient quality control measures are critical.

Software Environment

The system's software environment includes a range of deep learning frameworks, libraries, and tools selected to support each phase of the defect detection workflow:

Deep Learning Framework: PyTorch was selected as the primary framework for this study due to its dynamic computation graph and support for tensor computations, neural network modeling, and integration with pretrained models. PyTorch's

flexibility enables easy experimentation with model configurations, making it well-suited for iterative testing and optimization of architectures like U-Net and DeepLabV3+.

Auxiliary Libraries: OpenCV is utilized for initial image preprocessing tasks, including resizing, color space conversion (e.g., from grayscale to RGB), and tensor transformation. For data manipulation and statistical analysis, libraries such as NumPy and Scikit-learn are incorporated, especially in the development and implementation of Mini Batch Dictionary Learning for anomaly detection. These libraries support essential data processing tasks, providing reliable and efficient handling of image data throughout the system.

Pretrained Models and Transfer Learning: VGG16, a pretrained convolutional neural network on ImageNet, is integrated into the system as a feature extractor. VGG16's convolutional layers are fine-tuned to capture complex features within high-resolution images, making it an effective tool for preliminary feature extraction prior to anomaly detection and segmentation tasks. This transfer learning approach enhances model accuracy by leveraging pre-existing knowledge, thereby reducing the need for extensive labeled datasets and improving generalization capabilities.

Model-Specific Configurations

Each machine learning model within the system is individually configured to align with the specific requirements of the defect detection tasks:

Mini Batch Dictionary Learning: Configured to operate on small data subsets, Mini Batch Dictionary Learning iteratively processes the dataset in manageable batches, optimizing memory usage and computational efficiency. This setup supports the scalability of the algorithm to large datasets, essential for unsupervised learning environments where labeled defect data is limited. Sparse coding through this model facilitates the detection of anomalies by creating a compact and informative representation of normal and defective patterns in metallic images.

Custom U-Net and DeepLabV3+ Configurations: The U-Net model is specifically designed for pixel-level segmentation, with parameters such as kernel size, number of filters, and activation functions optimized for the segmentation of metallic surfaces. DeepLabV3+ is configured with Atrous Spatial Pyramid Pooling (ASPP) and varied dilation rates to enable multiscale feature extraction. This configuration allows DeepLabV3+ to capture fine details and complex textures, improving its accuracy in segmenting intricate defect regions. Both models are trained with high-resolution input settings to preserve spatial detail and ensure precise segmentation of defect areas.

Operational Considerations

The system is architected for modular deployment, allowing each component—Mini Batch Dictionary Learning, U-Net, and DeepLabV3+—to function independently or in conjunction, depending on the specific defect detection requirements. This modular approach provides flexibility, enabling the system to adapt to diverse types of metallic products and defect complexities. Such configurational adaptability is essential for real-time quality control in dynamic manufacturing settings, where various product lines may demand specific detection and segmentation strategies.

This configuration strategically combines advanced hardware and software resources, creating a robust environment capable of handling the complex, high-dimensional data involved in metallic defect detection. By optimizing both hardware efficiency and software versatility, the system achieves a balance between high-performance processing and precise defect identification, aligning with the goals of real-time industrial quality assurance.

3.2 Defect detection using Mini Batch Dictionary Learning Algorithm

In this section defect detection on metallic products is discussed in detail. Starting from the data collection, data preprocessing, application of Mini Batch Dictionary Learning with Sparse Coder model is explained in this chapter.

3.2.1 Data Collection

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). Description of the datasets are shown in Table 3.1.

Table 3. 1: MVTec Anomaly Detection Datasets [9]

Objects	Train	Test		Defect Groups	Image Dimension
	Good	Good	Defective		
Cable	224	58	92	8	1024 x 1024
Grid	264	21	57	5	1024 x 1024
Metal Nut	220	22	93	4	700 x 700
Screw	320	41	119	5	1024 x 1024
Transistor	213	60	40	4	1024 x 1024
Zipper	240	32	119	7	1024 x 1024
Total	1481	754			

3.2.2 Mini Batch Dictionary Learning Algorithm

Dictionary learning involves finding an optimal set of basic functions (dictionary) to represent data sparsely and efficiently. It is widely applied in fields like image processing and signal processing, aiming to minimize reconstruction error while achieving sparsity in representation.

So, what's Mini Batch Dictionary Learning with the same functionality of dictionary learning that offers computational advantages over batch-based approaches, particularly when dealing with large datasets.

It works in the following manner:

- A dictionary is created containing atoms, which serve as the fundamental building blocks for representing the data.
- Rather than processing the entire dataset in one go, Mini Batch Dictionary Learning operates on small random subsets, or mini-batches, of the data at each iteration. This approach significantly reduces computational overhead, enabling the algorithm to scale efficiently to large datasets.

Mini Batch Dictionary Learning efficiently learns dictionaries for sparse representation by iteratively updating them based on mini-batches of the dataset. Its ability to handle large datasets and potentially escape local minima [50].

3.2.3 Sparse Coder

When an image has been provided and a pre-learned dictionary is created to Sparse Coder to compute a sparse representation of the image using the dictionary's atoms. This involves finding coefficients that best approximate the image with sparse contributions from the dictionary. The output is a compressed representation of the image, retaining only the most important features. This process allows for efficient representation of new data without the computational overhead of dictionary learning.

3.2.4 VGG16

The VGG16 is a pre-trained CNN model which is used for object detection. Basically, this pre-trained model is used to extract features from the images in the datasets and this extracted feature was used to train Mini Batch Dictionary Learning model.

VGG16 is a popular convolutional neural network (CNN) model designed for image recognition and classification tasks. Its architecture is relatively simple, consisting of 16 layers with trainable weights (hence the name). These layers include:

- 13 convolutional layers
- fully connected layers

- Max-pooling layers in between
- SoftMax at the output for classification

The key idea behind VGG16 is the use of small 3x3 convolution filters throughout the network, which makes it deeper but with fewer parameters in each convolutional layer. This approach enhances the network's ability to learn intricate features without requiring excessively large filters or hyper-parameter tuning.

The architecture has proven effective for tasks like object detection and segmentation, and it's known for achieving strong results on image classification benchmarks such as ImageNet.

Despite being relatively simple compared to later models like ResNet or EfficientNet, VGG16 remains a go-to architecture for transfer learning, where a pre-trained model is fine-tuned for specific tasks [51].

3.2.4.1 VGG16 Architecture

A convolutional neural network model called VGG16 is employed in image recognition. It is distinct in that it relies on a small number of hyper-parameters, with only 16 layers having weights. Table 3.2 gives a summary of VGG16 architecture and Figure 3.1 depicts the VGG16 layers.

Table 3. 2:VGG16 Architecture Summary [52]

Layer	Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	224 x 224 x 3	-	-	-
1	2 X Convolution	224 x 224 x 64	3x3	1	relu
2	Max Pooling	112 x 112 x 64	3x3	2	relu
3	2 X Convolution	112 x 112 x 128	3x3	1	relu
4	Max Pooling	56 x 56 x 128	3x3	2	relu
5	2 X Convolution	56 x 56 x 256	3x3	1	relu

6	Max Pooling	28 x 28 x 256	3x3	2	relu
7	3 X Convolution	28 x 28 x 512	3x3	1	relu
8	Max Pooling	14 x 14 x 512	3x3	2	relu
9	3 X Convolution	14 x 14 x 512	3x3	1	relu
10	Max Pooling	7 x 7 x 512	3x3	2	relu
13	FC	25088	-	-	relu
14	FC	4096	-	-	relu
15	FC	4096	-	-	relu
Output	FC	1000	-	-	Softmax

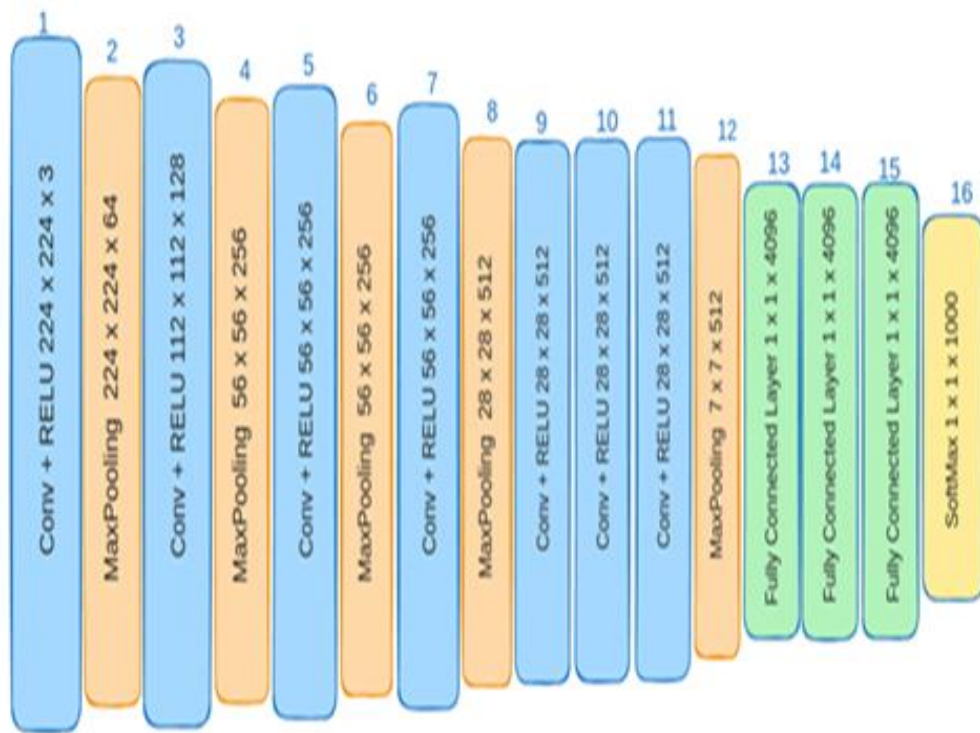


Figure 3. 1: VGG 16 layers

3.2.5 Methodology of Anomaly Detection

The anomaly detection methodology implemented in this study utilizes an unsupervised learning approach, specifically through Mini Batch Dictionary Learning with Sparse Coding. This method is designed to identify anomalies in metallic product images by differentiating between typical (non-defective) and atypical (defective) patterns. The process is comprised of several critical stages that ensure accurate detection of anomalies without reliance on labeled defect data.

1. Data Preparation

The initial phase involves preprocessing the images within the dataset to ensure consistency and compatibility with the anomaly detection model. Each image is resized, normalized, and converted into a format suitable for tensor manipulation, enhancing the model's ability to process high-resolution images typical of metallic products. This preprocessing step is essential for minimizing noise and standardizing input data across various metallic product categories.

2. Dictionary Learning

Dictionary learning is performed on non-defective images to capture a baseline representation of normal metallic surfaces. The method builds a sparse dictionary, or a set of features “atoms,” that encapsulates the fundamental characteristics of defect-free images. By training the model solely on normal images, it constructs a reference framework of what constitutes typical, acceptable surface characteristics for each type of metallic product. Sparse coding is applied to these normal images, optimizing for a compact and efficient representation that emphasizes the core structural and textural features of the product surfaces.

3. Anomaly Detection through Reconstruction Error

During the testing phase, new images are projected onto the learned dictionary to assess their conformity to the established normal patterns. If an image can be reconstructed within a low error margin using the learned atoms, it is classified as normal. However, images that exhibit high reconstruction errors—indicating that they

cannot be adequately represented by the normal dictionary—are flagged as anomalies. This approach leverages reconstruction error as the primary criterion for anomaly detection, where higher error values correspond to a greater likelihood of the image containing a defect.

4. Sparse Representation and Efficiency

The use of sparse representation in Mini Batch Dictionary Learning is critical for maintaining computational efficiency, especially given the high volume of images and the real-time requirements of defect detection in industrial settings. By processing data in mini-batches, the model minimizes memory usage and computational load, making it scalable for industrial applications. Additionally, sparse coding enables a focused representation by only retaining the most significant features of each image, which improves the model's precision in detecting subtle defects.

5. Post-Detection Categorization

After anomalies are detected, further processing can be applied to categorize defect types, such as "scratched," "bent," or "missing component." Although this stage is optional in unsupervised anomaly detection, it is valuable in industrial contexts where specific defect identification aids in quality control and decision-making. Clustering techniques or rule-based systems can be employed to group similar anomalies and assign defect labels, providing more detailed information about the nature of each detected defect.

This methodology is advantageous in scenarios where obtaining labeled defect data is impractical or costly, as it relies solely on learning from normal, defect-free images. By detecting deviations from normal patterns, the unsupervised approach effectively identifies defects while maintaining high adaptability and scalability in dynamic industrial environments.

3.2.5.1 Training the Model

Some steps are forwarded for training the model with dataset and finding the dictionary as shown in flowchart in Figure 3.2.

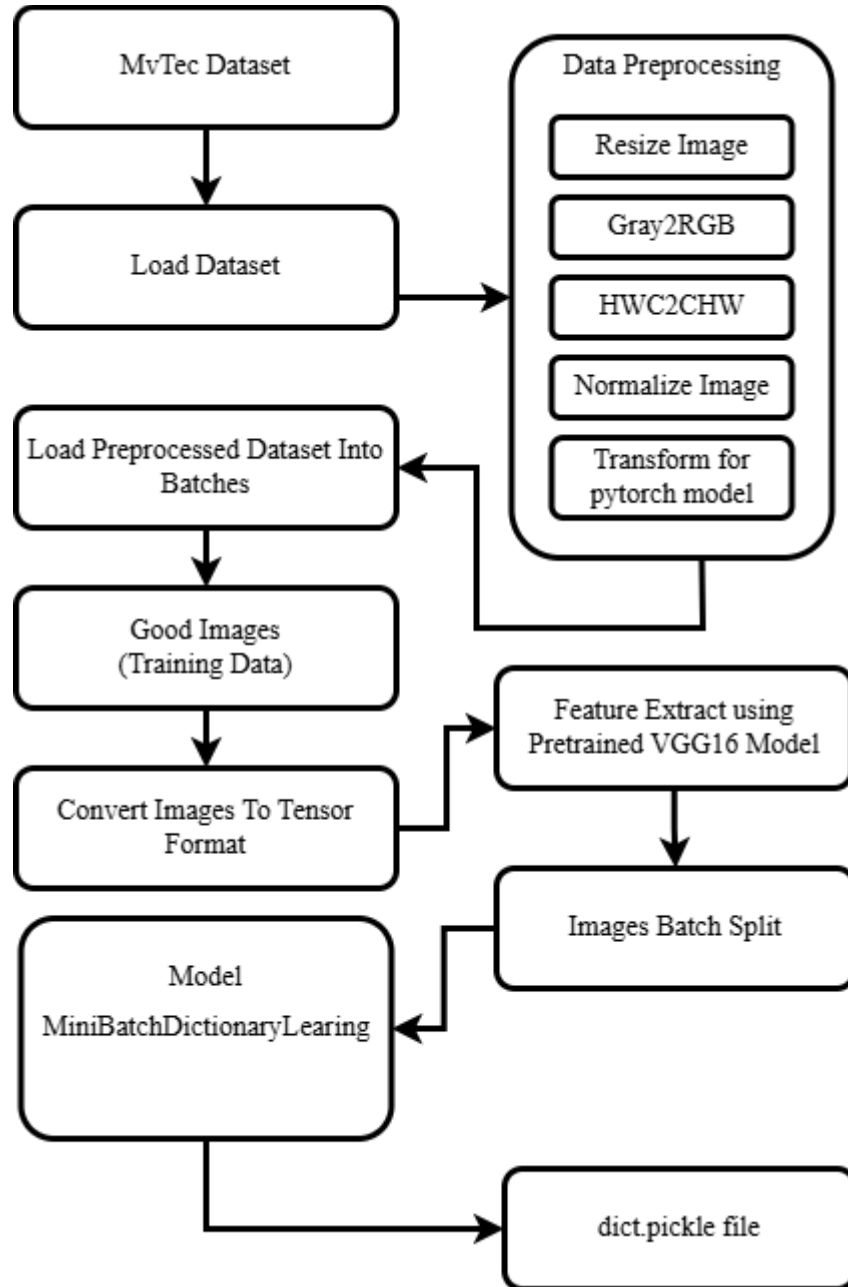


Figure 3. 2: Methodology for training purpose

1. Load Dataset: The first step involves loading the dataset into the system. This dataset could include a collection of images used to train the model.

2. Dataset Preprocessing: Once loaded, the dataset undergoes preprocessing steps to prepare it for further analysis. This can involve several techniques like:

- **Image Resizing:** Images are resized to a standard dimension.
- **Gray2RGB:** If the images are in grayscale format, they are converted to RGB (Red, Green, Blue) format, which uses three channels to represent color.
- **HWC2CHW:** This formatting step likely refers to changing the data layout from Height, Width, Channel (HWC) to Channel, Height, Width (CHW). This is a common data format used in deep learning frameworks like PyTorch.
- **Image Normalization:** After preprocessing, the images are normalized. This step scales the pixel values of each image to a specific range, often between 0 and 1. This helps to improve the training speed and performance of the machine learning model.
- **Convert Images to Tensor Format:** The images are then converted into tensor format, a multidimensional data structure commonly used in deep learning.
- **Transform for Torch Model:** The data is then transformed for use with a PyTorch model, a popular deep learning framework.

3. Extract Feature using Pretrained VGG16 Model: A pre-trained VGG16 model is used to extract features from the images. A pre-trained model is a machine learning model that has already been trained on a large dataset of images. This allows the current model to leverage the knowledge learned from the pre-trained model. Here this study used the transfer learning concept.

4. Images Batch Split: The data is then split into batches for training. This is done to improve memory efficiency and speed up the training process.

5. Mini Batch Dictionary Learning: Finally, a minibatch dictionary learning algorithm is used to train the model. This involves learning a set of basis vectors (dictionary) that can be used to represent the data efficiently. The image shows a methodology for an anomaly detection method with the minibatch dictionary learning system involves loading the dataset, preprocessing the data, normalizing the images, converting them to tensor format, transforming them for a PyTorch model, extracting

features using a pre-trained VGG16 model, splitting the images into batches, and finally using a minibatch dictionary learning algorithm to train the model.

3.2.5.2 Testing the Model

The methodology for testing the model with random data are shown with the flowchart as shown in Figure 3.3.

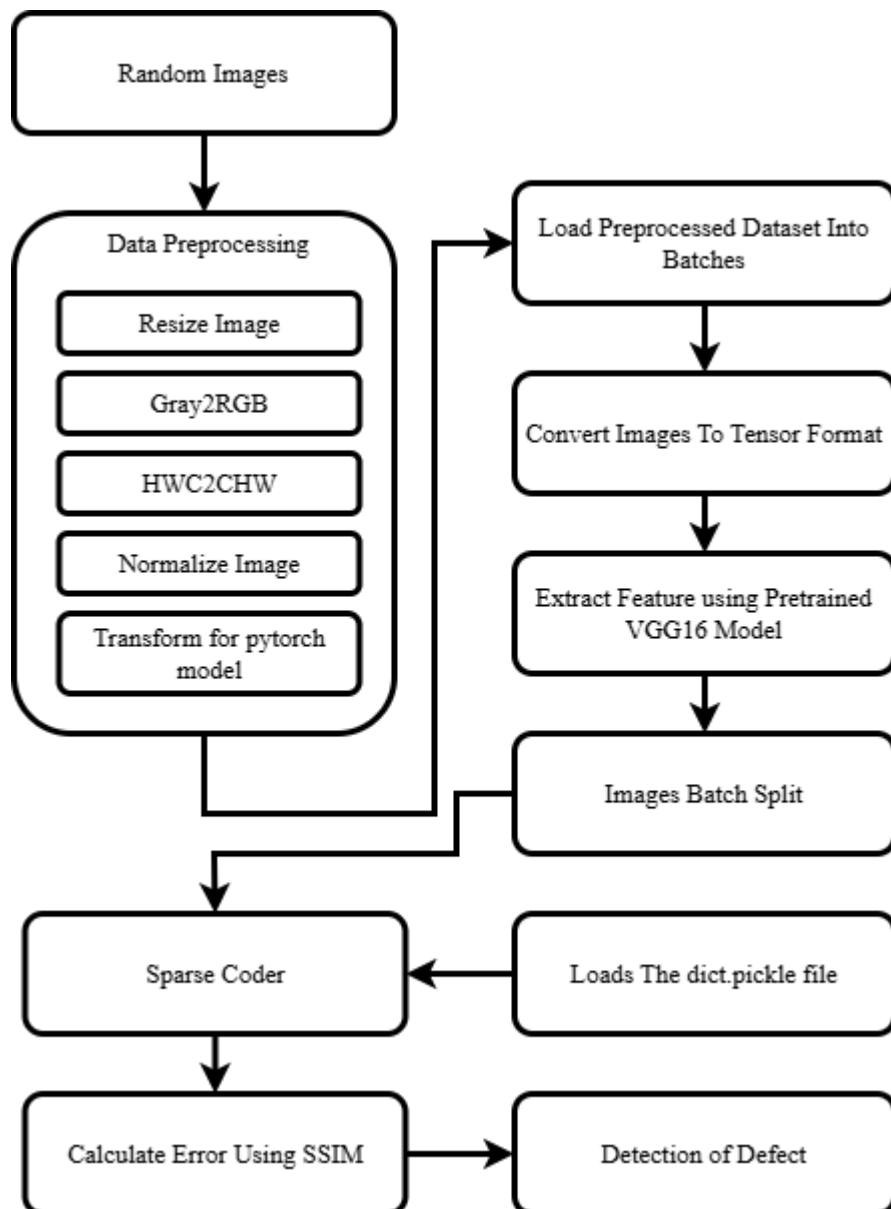


Figure 3. 3: Methodology applied for testing purpose

For the random image the pre-processing steps are the same until it passed to Sparse Coder.

1. Sparse Coding: The flowchart mentions a step that loads a dictionary file (pickle) the pre-trained dictionary is created using the Mini Batch Dictionary Learning in the training step. In sparse coding, a dictionary is a collection of basis vectors or atoms that are used to represent the input image. The image is represented as a sparse linear combination of these basis vectors. This step likely involves using a sparse coding algorithm to represent the extracted features from the previous step using the loaded dictionary.

2. Error Calculation: The sparse codes are then used to calculate the reconstruction error of the original image using the SSIM (Structural Similarity Index) metric. SSIM is a way of measuring the perceived quality of a reconstructed image compared to the original.

3. Defect Detection: Presumably, a threshold is applied to the reconstruction error to identify whether an image contains a defect. Images with a reconstruction error exceeding the threshold are likely to be classified as defective.

The flowchart outlines a process for automating defect detection in images. By using a pre-trained dictionary and sparse coding techniques, the system can learn to represent normal images efficiently and flag images with reconstruction errors exceeding a certain threshold as defective.

3.3 Defect segmentation on metallic products using Custom U-net model

Defect segmentation in metallic products is a critical task in industrial manufacturing, particularly for quality control and defect detection. The process involves identifying and segmenting defective areas in various metallic objects, such as grids, cables, metal nuts, screws, zippers, and transistors. This procedure typically utilizes image processing techniques and deep learning models like U-Net to enhance precision and efficiency in detecting anomalies. Below, more detailed explanation are presented for each component of the process.

3.3.1 Data Processing

The initial phase of defect segmentation focuses on data preparation and processing to ensure that the model can effectively learn from the input images. This begins by organizing the paths of both the actual images of metallic products and their corresponding mask images. The masks are essentially binary images where defective

areas are highlighted, and these are crucial for training a model to learn the distinction between defect-free and defective regions.

1. **Image and Mask Preparation:** The paths to the actual images and their corresponding mask images are loaded into two separate lists. This ensures that each image has its corresponding mask for training.
2. **Image Resizing:** Both the actual images and their masks are resized to a uniform size, typically 256x256 pixels. This step ensures consistency in the input dimensions, as machine learning models generally require uniform-sized inputs for optimal performance.
3. **Normalization:** After resizing, the pixel values of the images are normalized. Normalization scales the pixel intensity values to a standard range (often between 0 and 1), which speeds up the training process and helps the model converge faster. This step is vital for improving the computational efficiency of the neural network during training.
4. **Mask Channel Conversion:** The mask images, which might initially be in a 3-channel RGB format, are converted into a 1-channel format (grayscale). This conversion reduces memory usage significantly because the model is only concerned with binary information—whether a pixel belongs to a defective area or not.
5. **Data Splitting:** After the image and mask processing is complete, the dataset is split into two parts: training and testing sets. The training set is used to teach the model to recognize defects, while the test set is reserved to evaluate the model's performance on unseen data, ensuring the model can generalize to new images.

3.3.2 Model Training

To achieve accurate defect detection, a customized U-Net model is employed. U-Net is a popular neural network architecture, particularly well-suited for image segmentation tasks because of its ability to capture fine details and spatial hierarchies in images. Architecture of U-Net model is shown in Figure 3.4.

1. **Custom U-Net Architecture:** A U-Net model is specifically created for the task of feature extraction from images of metallic objects. This architecture is designed to extract hierarchical features from the images, which helps in detecting and localizing defects effectively. The U-Net model consists of an encoder-decoder structure where the encoder extracts features, and the decoder reconstructs the image's spatial details, making it ideal for segmentation tasks.

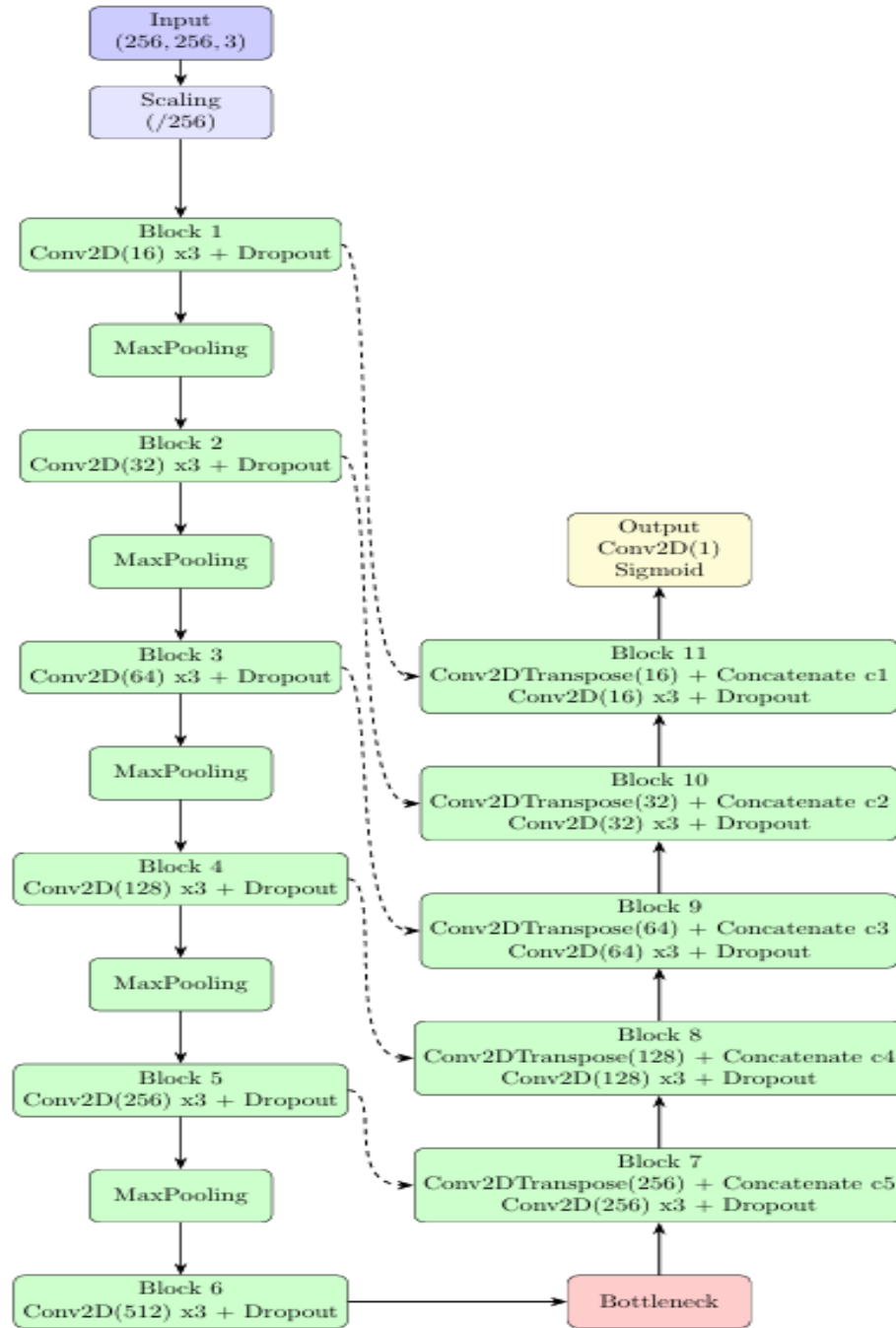


Figure 3. 4: Architecture of U-Net model

2. **Model Input and Output:** The model takes an actual image of a metallic object as its input, along with its corresponding mask, during training. The goal is to predict a binary mask that highlights the defective areas in the image. Essentially, the model learns to map from the actual image to the mask, which represents the defective regions. Training flowchart of custom U-Net model is shown in Figure 3.5.

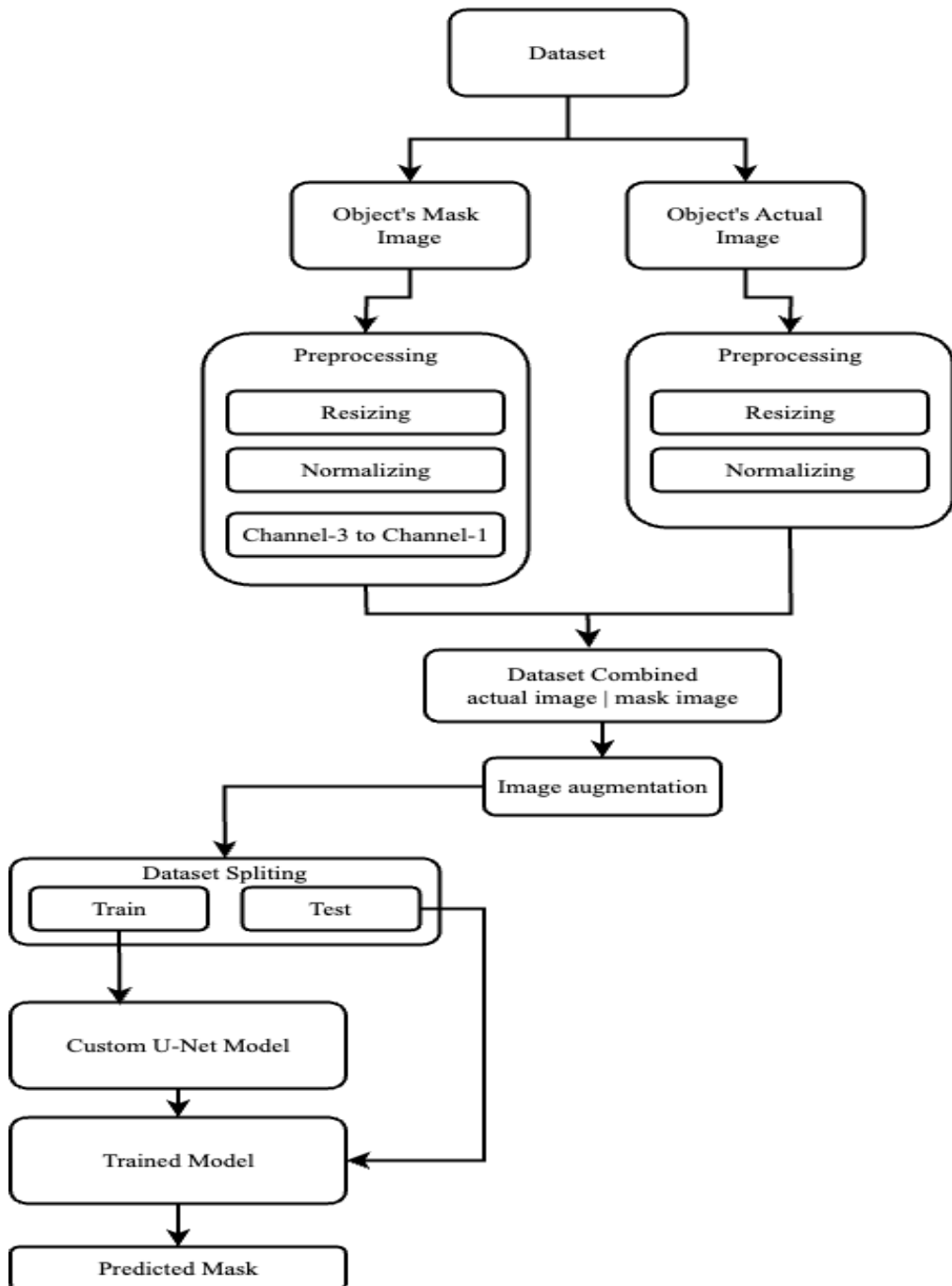


Figure 3. 5: Training flowchart of custom U-Net model

Defect Detection and Segmentation

Segmentation flowchart of Custom U-Net model is shown in Figure 3.6.

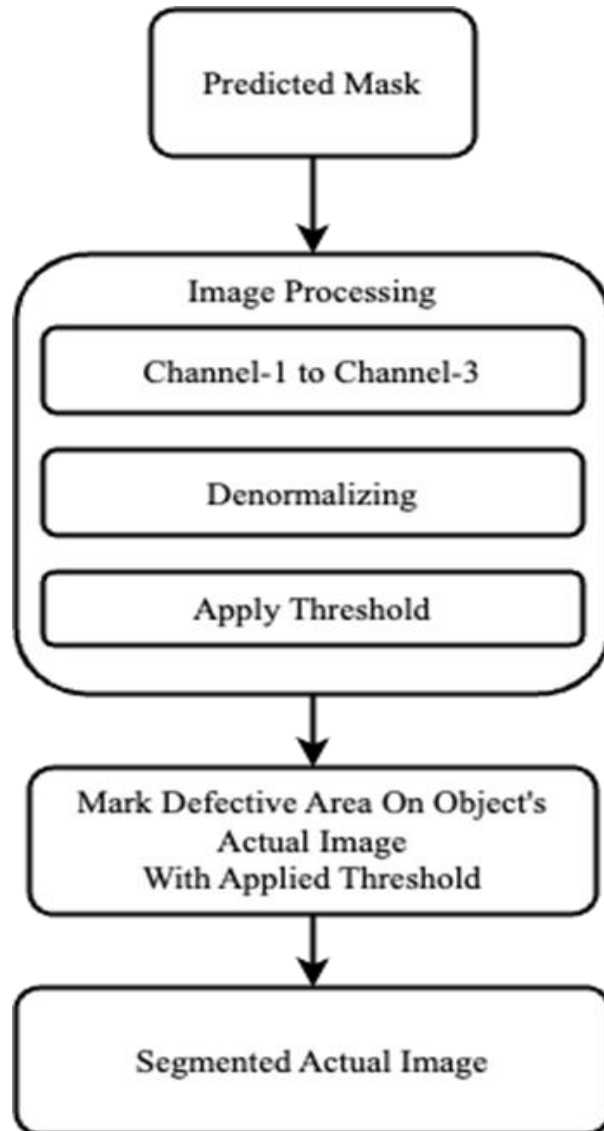


Figure 3. 6: Segmentation flowchart of Custom U-Net model

Once the U-Net model has been trained, it can be used to predict defective areas on metallic objects. This involves several steps, from loading the trained models to applying them to new images for defect detection.

1. **Model Saving for Multiple Objects:** After completing the training process, the trained model is saved for each of the six metallic objects: grids, cables,

metal nuts, screws, zippers, and transistors. This ensures that specific models can be used for defect detection on different object types, as each object might have distinct defect patterns.

2. **Defect Prediction (Mask Generation):** For defect detection, a new image (either defective or non-defective) is passed through the trained model. The model then generates a predicted mask. This predicted mask is in a resized, normalized, and 1-channel format, similar to the masks used during training. The mask highlights the defective areas in the image, making it easier to identify potential defects.
3. **Identifying Defective Areas:** The predicted mask can be visualized to see where the model has identified defects. This step allows inspectors or automated systems to quickly determine whether a product is defective and pinpoint the exact regions where the defects are located.

3.3.3 Segmentation Process

Once the defects have been detected, the next step is segmentation, where the defective areas are isolated and visualized on the original image.

1. **Denormalization:** After the mask prediction, the predicted mask is first denormalized. This step converts the pixel values of the mask back to their original range (e.g., from 0-1 back to 0-255), making it easier to interpret visually.
2. **Channel Conversion:** The predicted mask is then converted from a 1-channel image back to a 3-channel RGB image. This step is necessary to align the mask with the actual image, allowing for effective segmentation and visualization of the defective areas.
3. **Defective Area Segmentation:** With the mask now in a 3-channel format, custom segmentation functions are used to overlay the defective areas onto the original image. These functions segment the defective regions, making it easier to analyze and visualize the defective parts of the metallic object.

Through this process, manufacturers can efficiently detect and segment defects in various metallic products, ensuring quality control and reducing the number of

defective products that make it to the market. This combination of data processing, model training, and defect segmentation provides a robust solution for industrial defect detection, offering both precision and scalability in handling large volumes of data.

3.4 Defect segmentation on metallic products using DeepLabV3+ model

DeepLabV3+ is an advanced semantic segmentation model as shown in Figure 3.7 that builds upon the foundations laid by its predecessor, DeepLabV3. It introduces an encoder-decoder architecture designed to enhance segmentation accuracy, particularly at object boundaries. Below is a detailed overview of its architecture and components[53], [54].

Key Components of DeepLabV3+ Architecture

- **Encoder:** The encoder is responsible for extracting rich semantic features from the input image. It typically uses a backbone network such as ResNet or Xception, which employs depth wise separable convolutions to reduce computational cost while maintaining performance [55].
- **Atrous Convolution:** DeepLabV3+ utilizes atrous (or dilated) convolution, which allows for a larger receptive field without increasing the number of parameters. This is crucial for capturing multi-scale contextual information effectively. The atrous rate determines the spacing between the kernel points, allowing the model to gather broader context from the input data [55].
- **Atrous Spatial Pyramid Pooling (ASPP):** This module applies multiple parallel atrous convolutions with different rates to capture features at various scales. By aggregating these features, ASPP enhances the model's ability to recognize objects of different sizes and improves overall segmentation accuracy [55].
- **Decoder:** The decoder refines the coarse segmentation results produced by the encoder. It combines high-level feature maps from the encoder with low-level features to improve localization and delineation of object boundaries. This step is essential for achieving precise pixel-wise predictions [55].

Architecture Structure

Architecture of DeepLabV3+ Model [56] is shown in Figure 3.7.

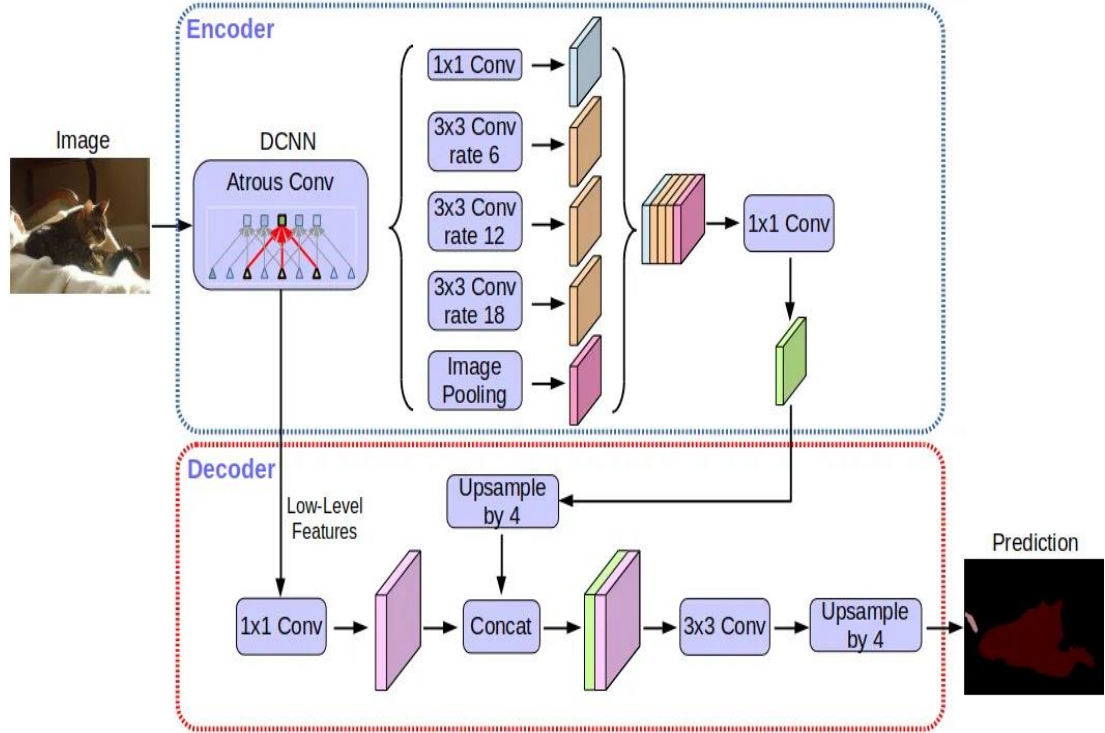


Figure 3. 7: Architecture of DeepLabV3+ Model[56]

1. **Input Layer:** The model accepts images of size $H \times W \times C$ (height, width, channels).
2. **Backbone Network:** Commonly ResNet (e.g., ResNet50 or ResNet101) or Xception. The backbone extracts hierarchical features through several convolutional layers and pooling operations.
3. **ASPP Module:** Applies Atrous convolutions with different dilation rates (e.g., 1, 6, 12, 18) to capture multi-scale features. Combines outputs from these convolutions into a single feature map.
4. **Decoder Module:** Upsamples the feature map from ASPP. Concatenates it with corresponding low-level features from earlier layers in the backbone to enhance spatial resolution. Further processes this combined feature map to refine segmentation results.
5. **Output Layer:** Produces a final segmentation map with pixel-wise class predictions.

3.5 Conclusion

In this chapter, we implemented machine learning techniques for detecting and segmenting defects in metallic products. The Mini Batch Dictionary Learning with Sparse Coder model proved effective for unsupervised anomaly detection, particularly with complex metallic objects. Additionally, the Custom U-Net model provided accurate pixel-level segmentation of surface defects, showing strong performance across various metallic objects. The chapter also highlighted the importance of integrating deep learning architectures like DeepLabV3+, which further enhanced the defect detection process by refining segmentation capabilities and improving accuracy. Together, these approaches form a comprehensive framework for automated industrial Defect detection, minimizing human error and improving overall product quality.

Chapter 04

Results and Discussion

4.1 Introduction

This chapter focuses on the results obtained from applying machine learning techniques for Defect detection and segmentation in metallic products. It showcases the performance of three key models: Mini Batch Dictionary Learning with Sparse Coder, Custom U-Net, and DeeplabV3+. The Mini Batch Dictionary Learning model demonstrated effective anomaly detection across various metallic objects like cables, grids, and screws. Meanwhile, the Custom U-Net model provided high-precision, pixel-level segmentation of defects, ensuring accurate identification of surface flaws. Lastly, the DeeplabV3+ model further enhanced segmentation accuracy, particularly in detecting subtle defects, and achieved impressive performance metrics. This chapter highlights the efficiency and applicability of these machine learning models in improving industrial quality control.

4.2 Results obtained for Defect detection on metallic products using Mini Batch Dictionary Learning with Sparse Coder

In this section, the results of the experiments evaluating the performance of the proposed model on the MVTec dataset are presented. The effectiveness of the approach in detecting anomalies in various categories of industrial objects was assessed. Our proposed model trained with 1481 good images from 6 different types of objects using Mini Batch Dictionary Learning model and tested with 754 random images contains defective and non-defective images using the pre-trained dictionary from the Mini Batch Dictionary Learning model and Sparse Coder and successfully detected the defective images.

4.2.1 Results obtained for anomaly detection (unsupervised approach) for cable using Mini Batch Dictionary Learning with Sparse Coder model

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of cable like missing wire defect, missing cable defect, cut cable inner insulation, combined defect are detected anomaly in Figure 4.1-Figure 4.4.



Figure 4. 1: Missing Wire defect of cable

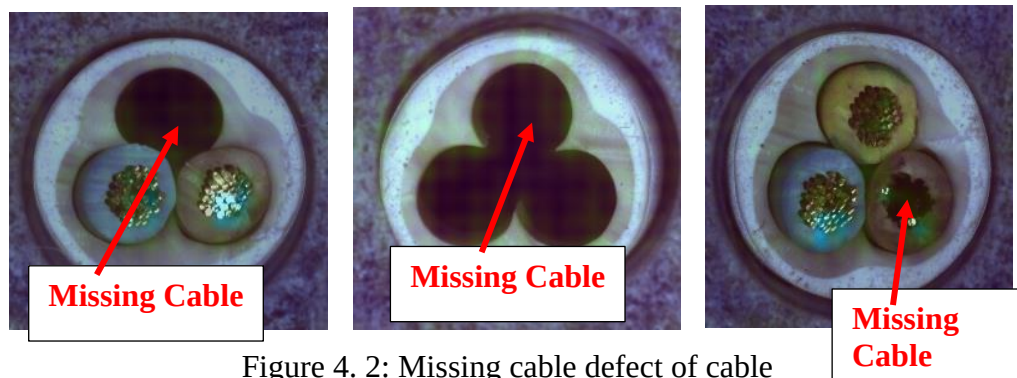


Figure 4. 2: Missing cable defect of cable

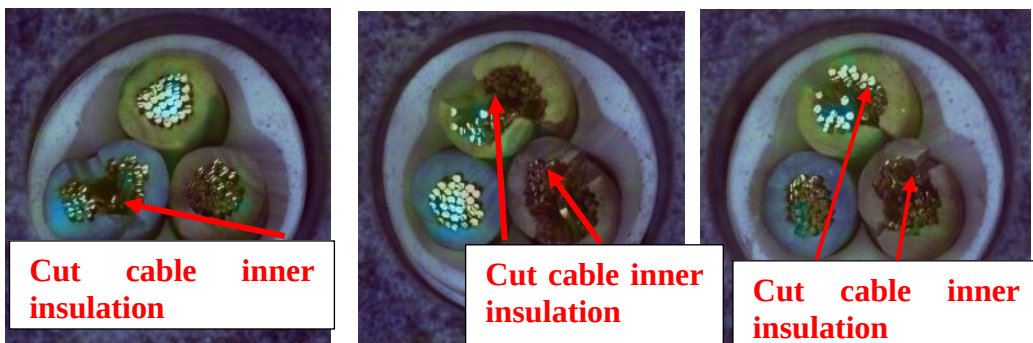


Figure 4. 3: Cut cable inner insulation defect of cable

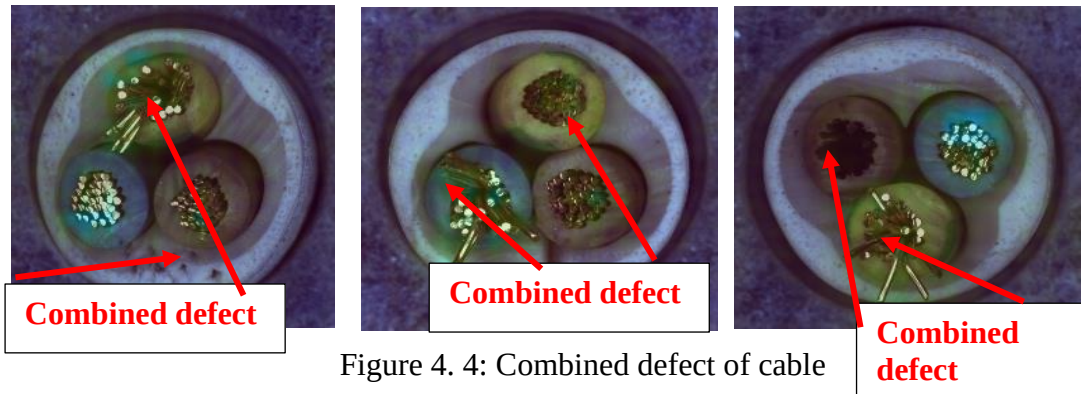


Figure 4. 4: Combined defect of cable

4.2.2 Results obtained for anomaly detection (unsupervised approach) for Grid using Mini Batch Dictionary Learning with Sparse Coder model

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of Grid like Bent, Broken, Glue, Metal contamination, Thread defect are detected anomaly in Figure 4.5-Figure 4.10.

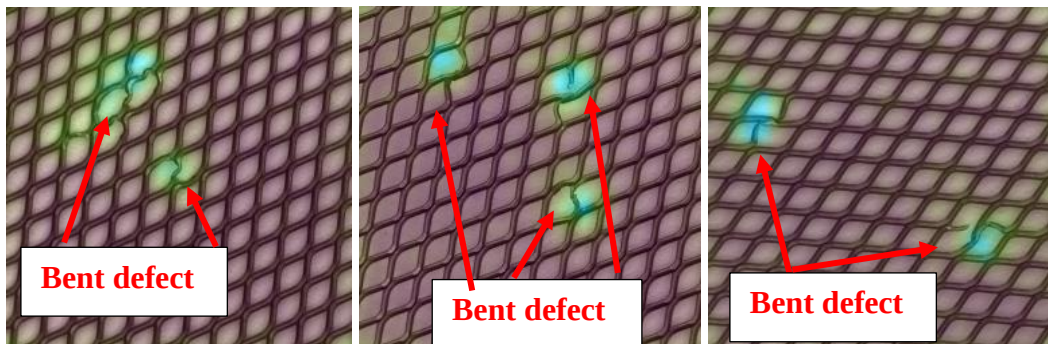


Figure 4. 5: Bent defect of Grid

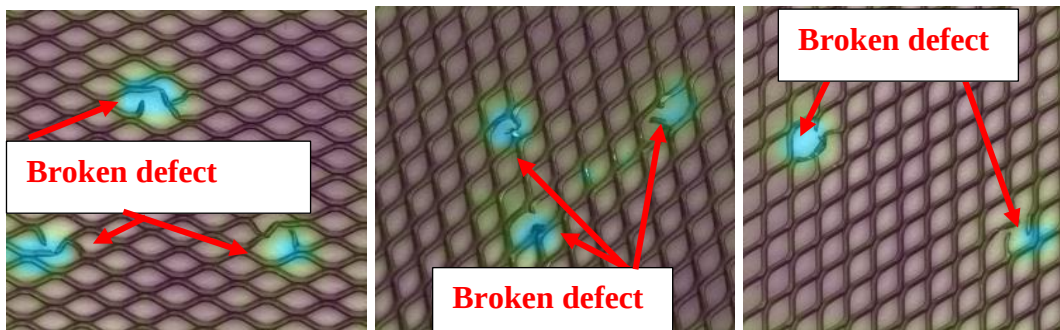


Figure 4. 6: Broken defect of Grid



Figure 4. 7: Glue defect of Grid

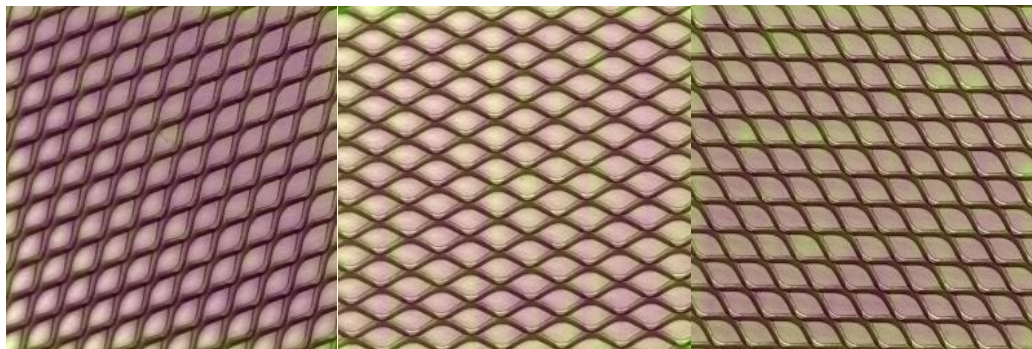


Figure 4. 8: Good Image of Grid

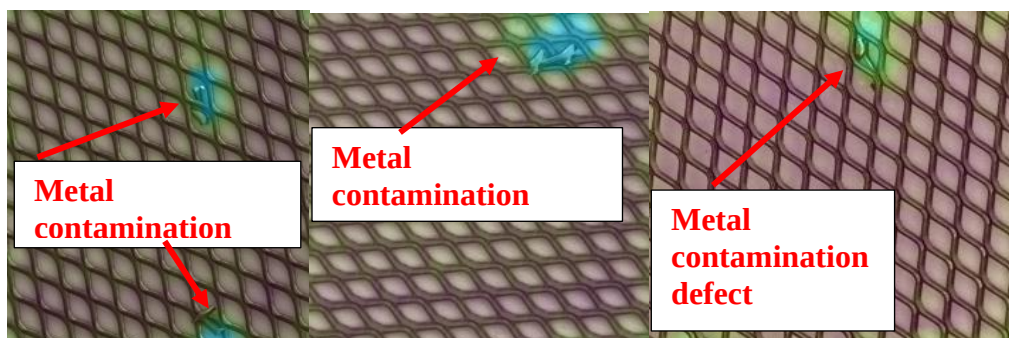


Figure 4. 9: Metal contamination defect of Grid

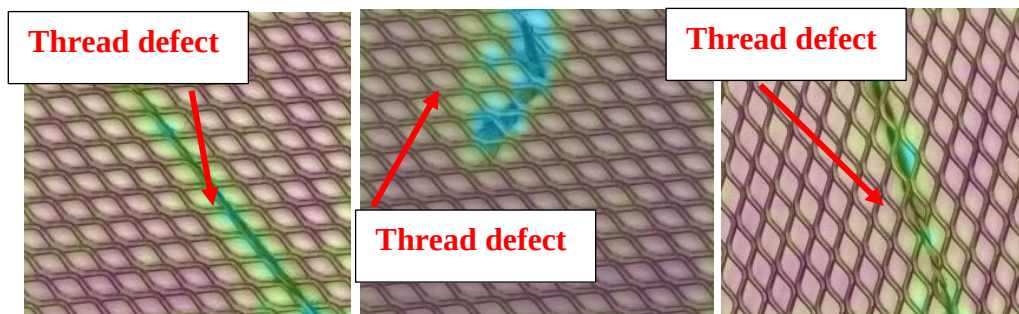


Figure 4. 10: Thread defect of Grid

4.2.3 Results obtained for anomaly detection (unsupervised approach) for Metal Nut using Mini Batch Dictionary Learning with Sparse Coder model

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of Metal Nut like bent, color defect, good image, flip defect are detected anomaly in Figure 4.11-Figure 4.14.

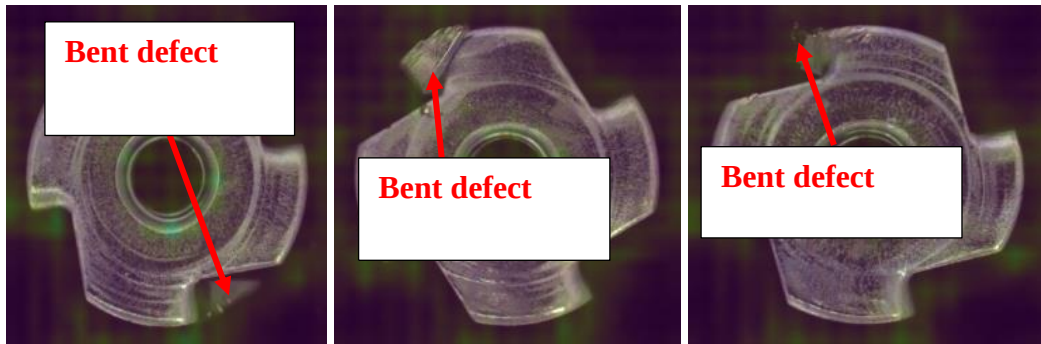


Figure 4. 11: Bent defect of metal nut

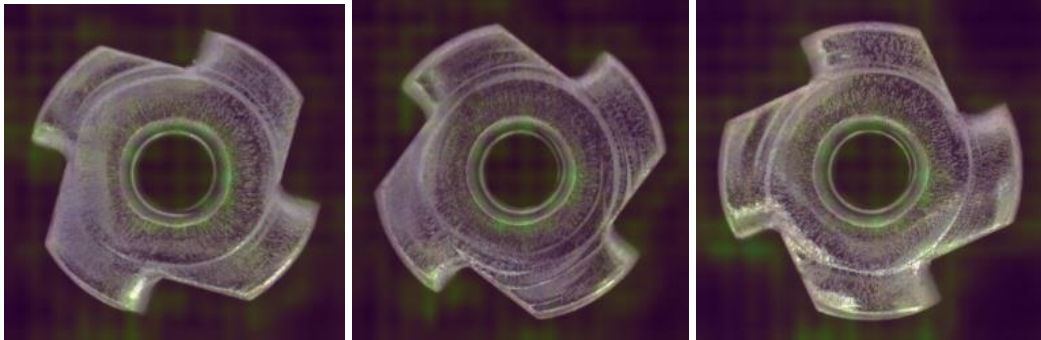


Figure 4. 12: Good image of metal nut

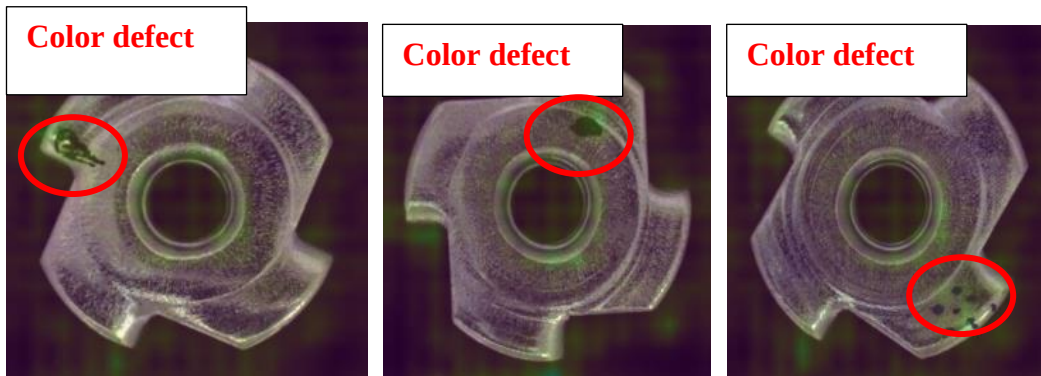


Figure 4. 13: Color defect of metal nut



Figure 4. 14: Flip defect of metal nut

4.2.4 Results obtained for anomaly detection (unsupervised approach) for Screw using Mini Batch Dictionary Learning with Sparse Coder model

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of Screw like manipulated front, scratch head, scratch neck, thread side, thread top defect are detected anomaly in Figure 4.15-Figure 4.19.

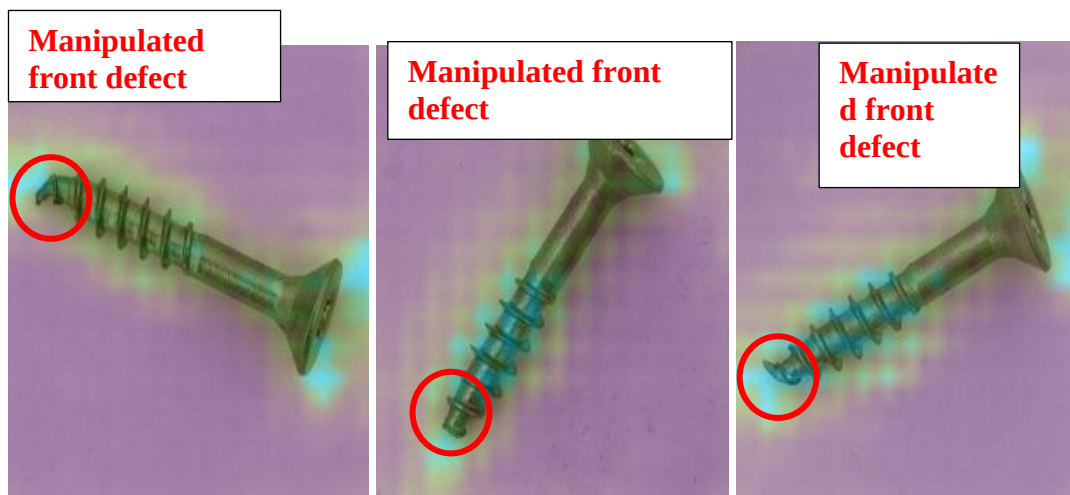


Figure 4. 15: Manipulated front defect of screw

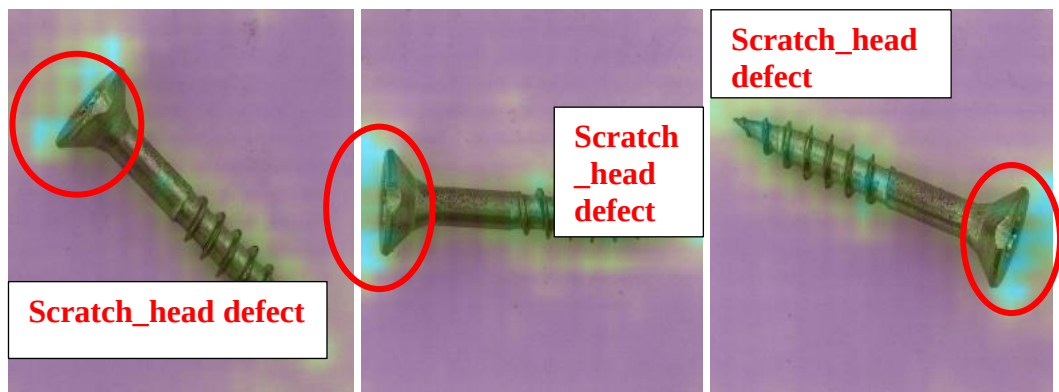


Figure 4. 16: Scratch head defect of screw

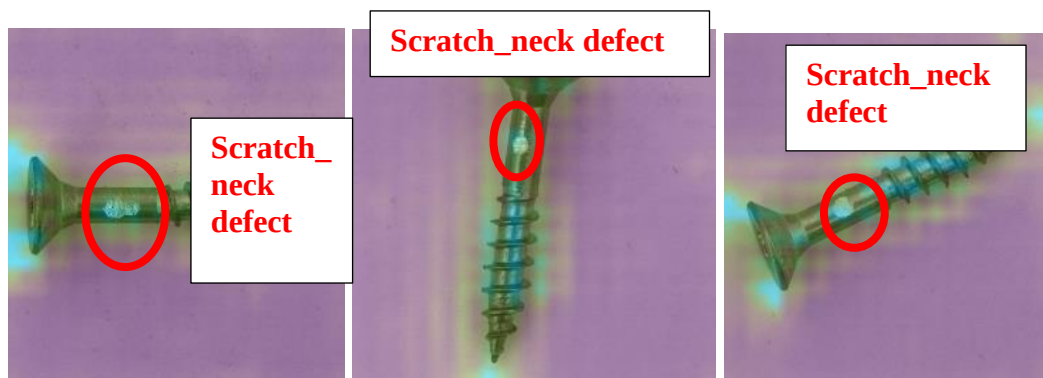


Figure 4. 17: Scratch_neck defect of screw

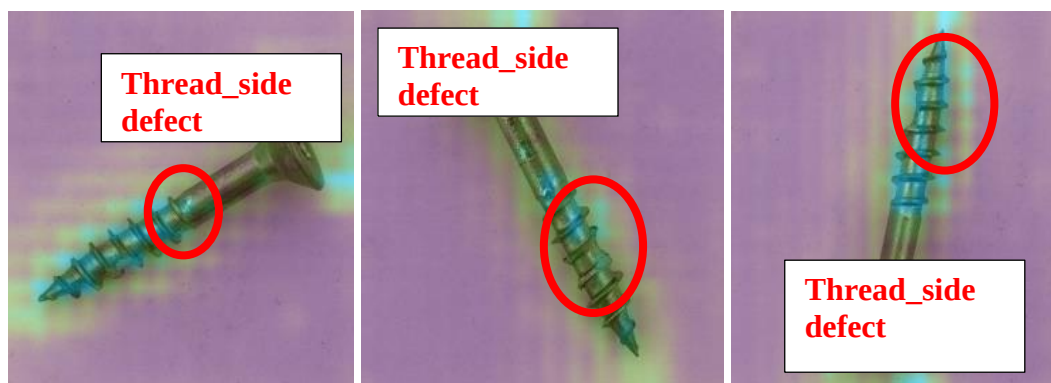


Figure 4. 18: Thread side defect of screw



Figure 4. 19: Thread top defect of screw

4.2.5 Results obtained for Defect detection and segmentation for Transistor using Mini Batch Dictionary Learning with Sparse Coder model

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of Transistor like bent lead, cut lead, misplaced, damaged case defect are detected anomaly in Figure 4.20-Figure 4.23.

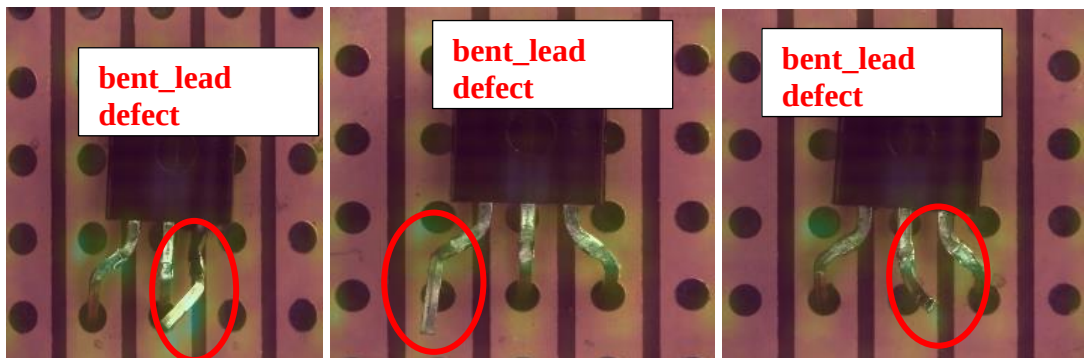


Figure 4. 20: bent_lead defect of transistor

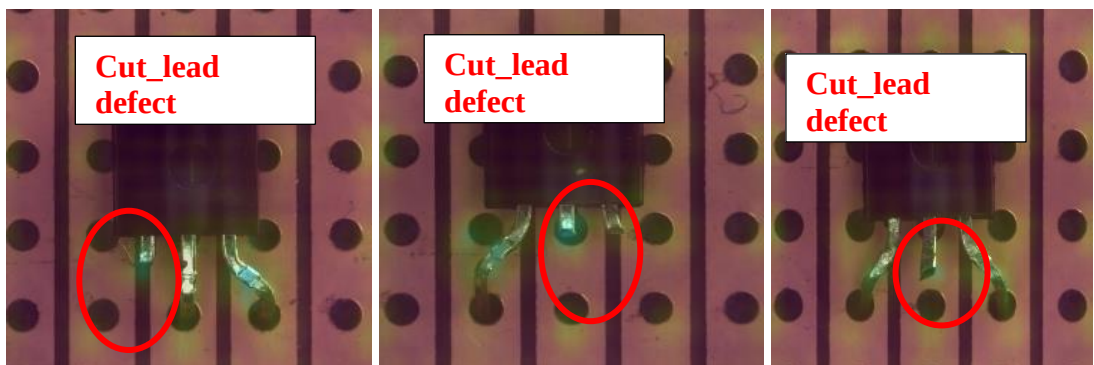


Figure 4. 21: Cut_lead defect of transistor

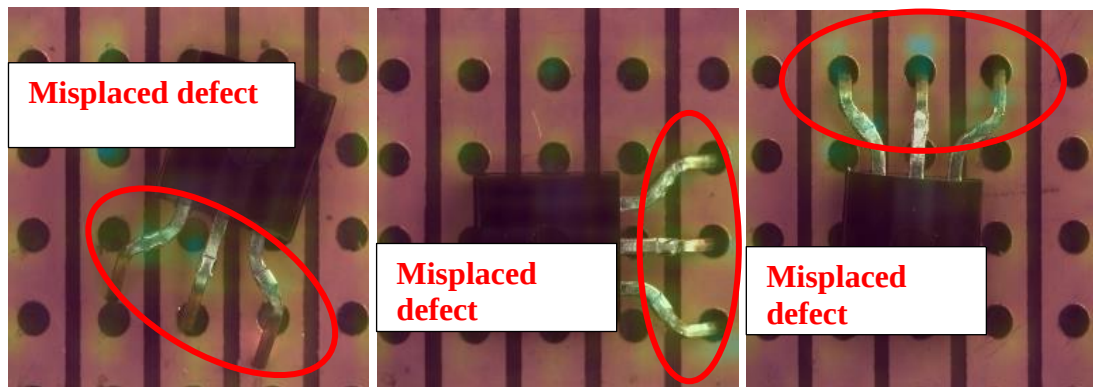


Figure 4. 22: Misplaced defect of transistor



Figure 4. 23: Damaged_case defect of transistor

4.2.6 Results obtained for Defect detection and segmentation for Zipper using Mini Batch Dictionary Learning with Sparse Coder model

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of Zipper broken teeth, combined defect, fabric border defect, fabric interior, rough defect, good image, split teeth defect are detected anomaly in Figure 4.24-Figure 4.30.

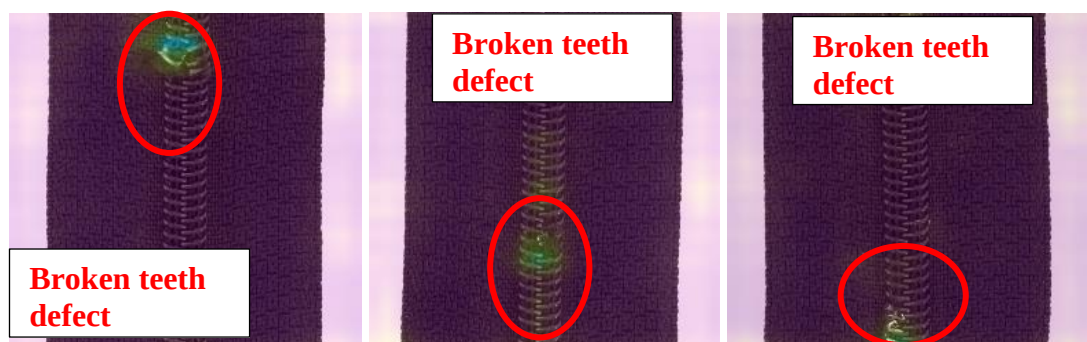


Figure 4. 24: Broken teeth defect of zipper

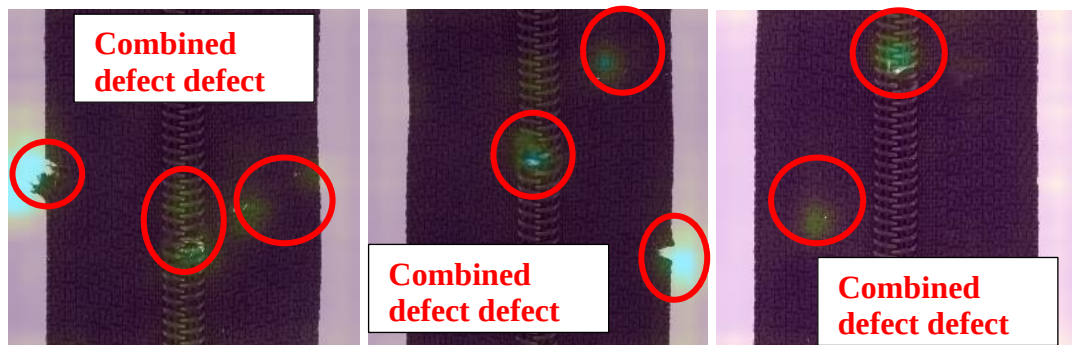


Figure 4. 25: Combined defect of zipper

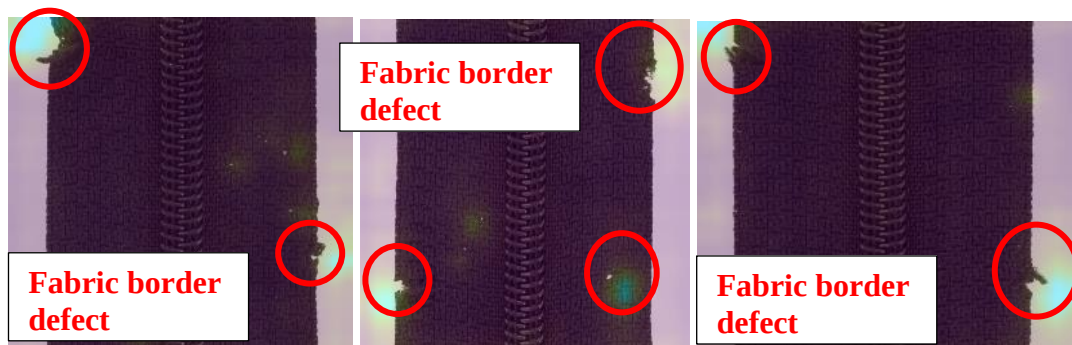


Figure 4. 26: Fabric border defect of zipper

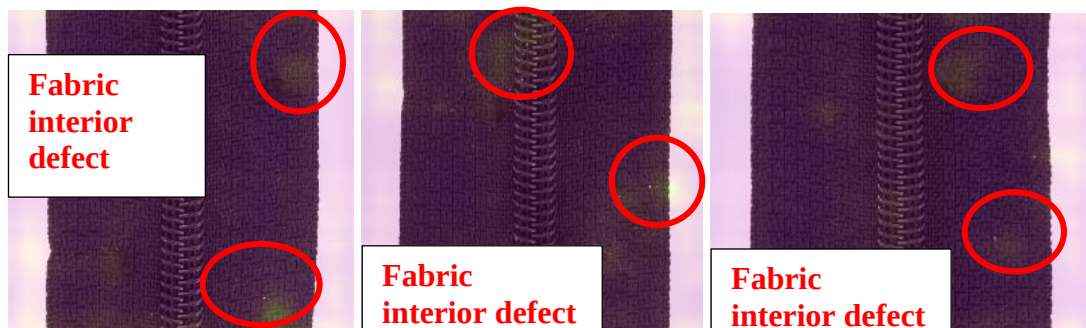


Figure 4. 27: Fabric interior defect of zipper



Figure 4. 28: Good image of zipper

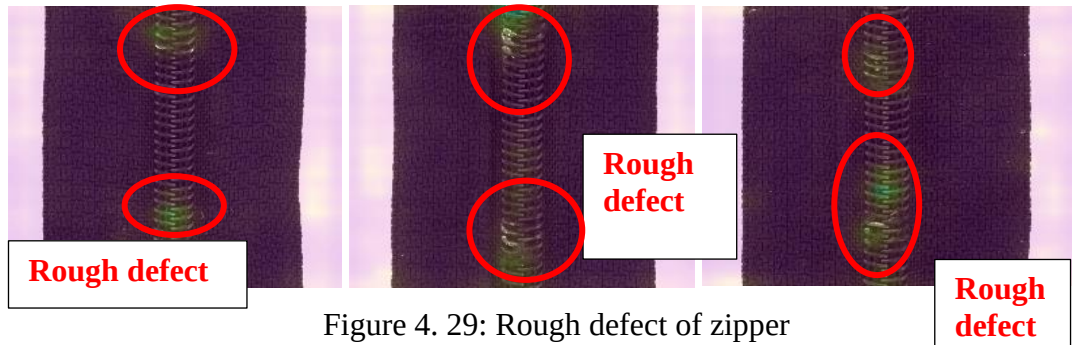


Figure 4. 29: Rough defect of zipper

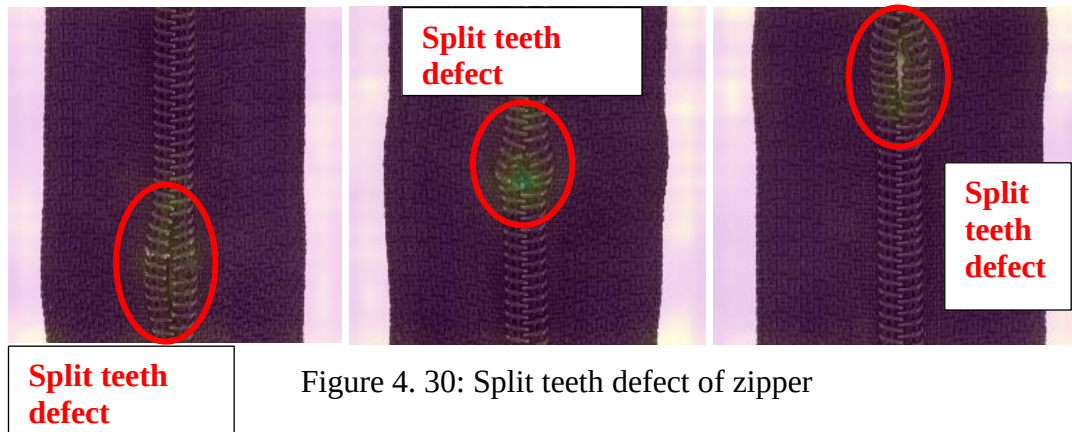


Figure 4. 30: Split teeth defect of zipper

On the MVTec dataset, promising results were obtained, proving the model's high precision anomaly detection. The experiments' main assessment criterion was Average Precision (AP).

A comparison of the results with those reported in the work of [2] is provided in Table 2. The Average Precision values reported in our study and those from the reference paper are compared and presented in Table 4.1.

Table 4. 1: Average Precision Recall for different objects

Algorithm	Object Type					
	Cable	Grid	Metal nut	Screw	Transistor	Zipper
f-AnoGAN	0.098	0.05	0.273	0.062	0.13	0.027
Feature dictionary	0.617	0.213	0.76	0.017	0.364	0.369
L2-autoEncoder	0.108	0.252	0.416	0.147	0.381	0.095

SSIM- autoEncoder	0.052	0.081	0.359	0.05	0.191	0.088
Proposed Model: Mini batch dictionary learning with Sparse coder	0.708	0.976	0.814	0.747	0.489	0.987

The top-performing solution for each dataset category is boldfaced. Our proposed methodology has been improved anomaly detection with respect to all other approaches.

Defect detection using Custom U-NET and DeepLabV3+:

The accuracy of the defect detection model, specifically using the DeepLabV3+ and U-Net architectures, varied depending on the dataset and model configuration. Here are key observations on the training iterations and accuracy based on the research:

1. DeepLabV3+ Model:

- For metallic defects, this model achieved high accuracy levels across various types of objects. For instance, while training on the “Grid” category, the DeepLabV3+ model reached an accuracy of 98.94% at approximately 50 epochs. Validation accuracy stabilized after the 10th epoch, with fluctuations indicating adjustments in generalization.

- For other objects, such as "Metal nut," the model performed optimally with an accuracy of 99.02% by epoch 50, showing consistent convergence with low error and indicating effective generalization in the learning process.

2. Custom U-Net Model:

- The U-Net architecture, particularly for defect segmentation in metallic products, reached around 98.79% accuracy after 50 epochs. Early convergence was observed around epoch 10, with minimal divergence between training and validation loss, indicating strong generalization.

- For smaller or intricate defects, such as those in "Zipper," the U-Net model achieved its best performance by epoch 46, showing a stable validation accuracy close to 99%, reflecting high precision in detecting smaller defects.

3. Training Observations Across Models:

- Both models showed rapid improvement within the initial epochs, typically reaching over 90% accuracy by epoch 5 and continuing to stabilize through later epochs. The best model performance varied slightly per object type but generally required around 50 epochs to achieve optimal accuracy with minimal loss.

In summary, training iterations around 50 epochs were generally sufficient for the model to achieve high accuracy, with DeepLabV3+ and Custom U-Net models each demonstrating excellent defect detection capabilities. The steady convergence of validation accuracy across iterations indicates that the models are well-suited for real-time defect detection applications in industrial environments.

4.3 Results obtained for Defect detection using segmentation on metallic products using Custom U-Net model

The outcomes of using machine learning approaches for flaw identification and segmentation in metallic products are the main topic of Chapter 4 of this thesis. High-precision, pixel-level defect segmentation was made possible by the Custom U-Net model, guaranteeing precise surface flaw identification.

4.3.1 Results obtained for Defect detection using segmentation for cable using Custom U-Net model

This section describes learning curve from the trained model for cable, accuracy of the trained model for cable, predicted mask for the trained model for cable and Defected image with the predicted mask applied for Cable.

4.3.1.1 Learning curve from the trained model for cable using Custom U-Net model

The Figure 4.31(a) illustrates the progression of the training and validation loss, while Figure 4.31(b) demonstrates the training and validation accuracy. Initially, the training loss decreases significantly, indicating that the model is learning and minimizing the error on the training set. The validation loss follows a similar trend, but with slightly higher values, showing generalization to unseen data. After around 10 epochs, both losses continue to decrease at a slower rate, stabilizing by the 50th epoch. The model achieves its best performance on the validation loss at epoch 49, marked by the blue dot, with a validation loss close to 0.1, signaling a well-fitting model without significant overfitting.

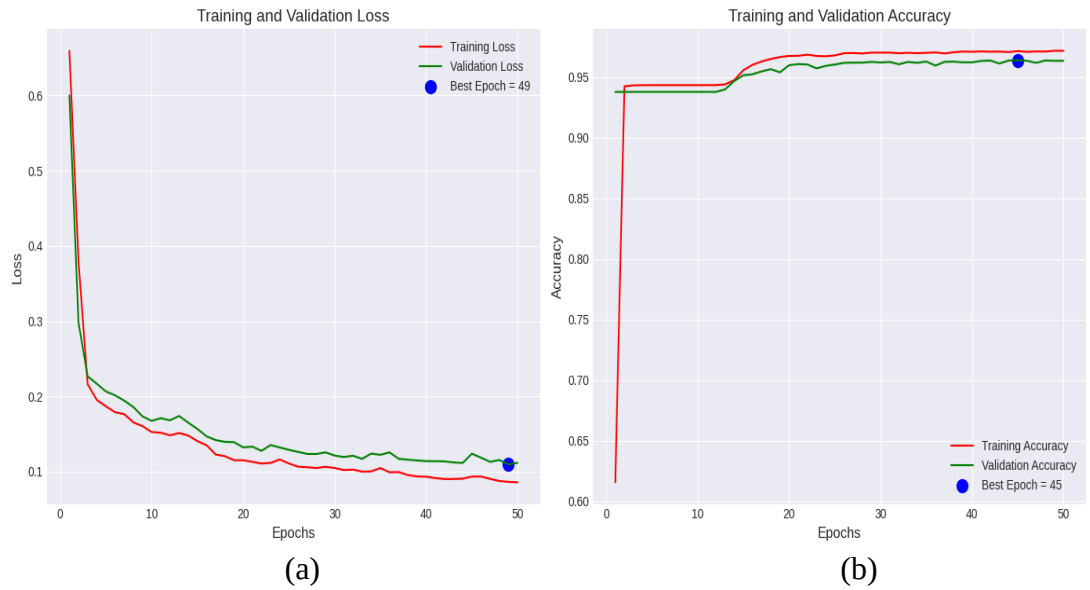


Figure 4. 31: Learning curve for cable using Custom U-Net model

Similarly, the accuracy curves in Figure 4.31(b) indicates that the model's accuracy rapidly improves in the initial epochs. By epoch 10, both the training and validation accuracies exceed 90%, continuing to climb and stabilize near 95% accuracy. The blue dot marks the best validation accuracy at epoch 45, indicating optimal generalization performance. The close proximity of training and validation accuracies, with minimal deviation, further suggests that the model is neither underfitting nor overfitting, successfully generalizing to unseen data while achieving high performance during training.

4.3.1.2 Accuracy of the trained model for cable

This figure 4.32 shows the evaluation results of a machine learning model, likely a neural network (based on the use of `model U-Net`), being tested on a dataset (`X`, `Y`). The key metrics are accuracy and loss:

1. Accuracy: The model achieved an accuracy of 0.9673 (or approximately 96.73%), meaning that nearly 97% of the predictions made by the model were correct when compared to the actual target values.

2. Loss: The loss value is 0.0933, indicating the error rate of the model's predictions. A lower loss value typically suggests better performance, especially if you're using a loss function like categorical cross-entropy.

These results show that the model is performing well, with high accuracy and a relatively low loss value. The evaluation took 16 seconds, processing 4 steps per second. This suggests the model has been trained effectively, but further tuning or validation might be necessary to ensure generalization across new data.

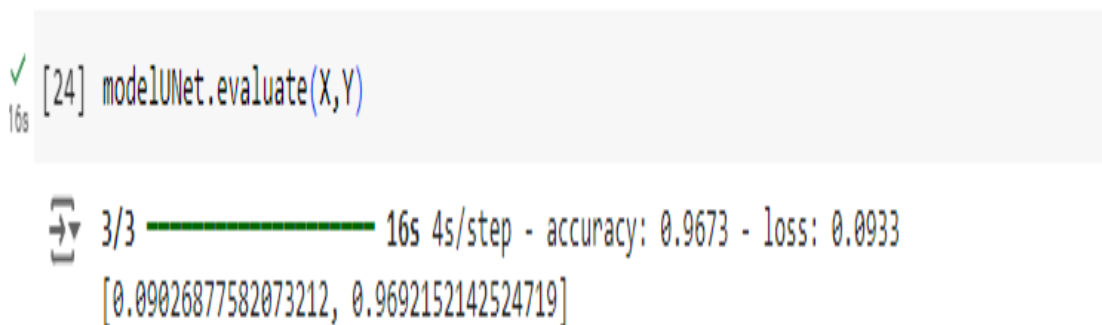


Figure 4. 32: Accuracy of the trained U-Net model for cable

4.3.1.3 Predicted mask for the trained model for cable

Figure 4.33 shows the predicted mask of the different defected area of the cable after training the model.

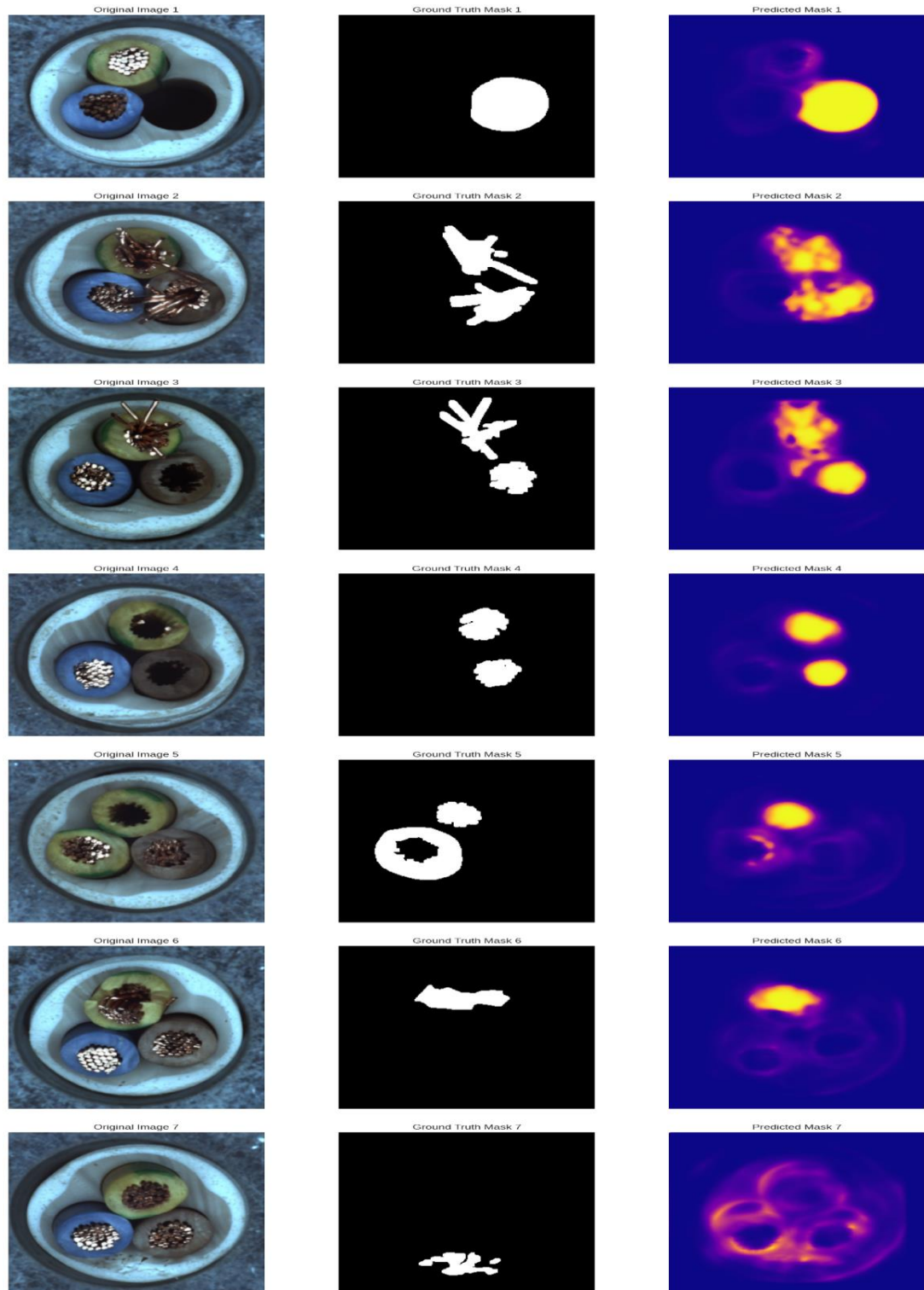


Figure 4. 33: Predicted mask for the trained U-Net model for cable

4.3.1.4 Defected image with the predicted mask applied for Cable

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of Cable like bent wire, cable swap, combined defect, cut inner insulation, cut outer insulation, missing cable and poke insulation defects are segmented in Figure 4.34-Figure 4.40.

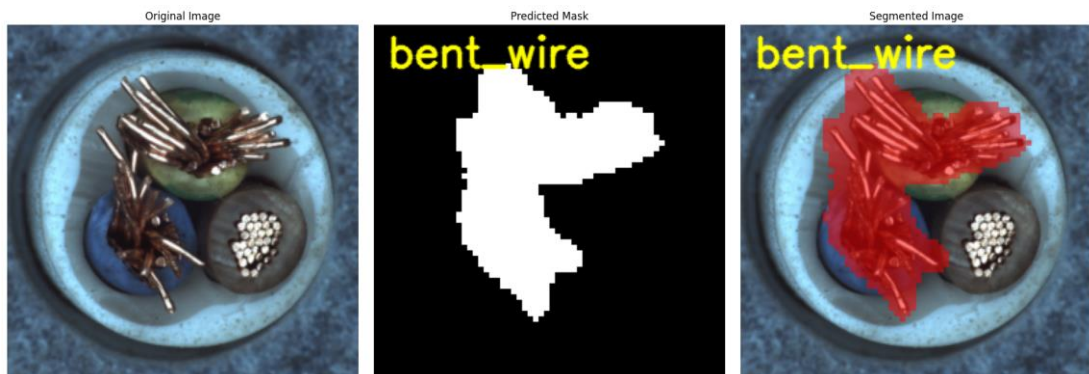


Figure 4. 34: Bent wire defect of cable

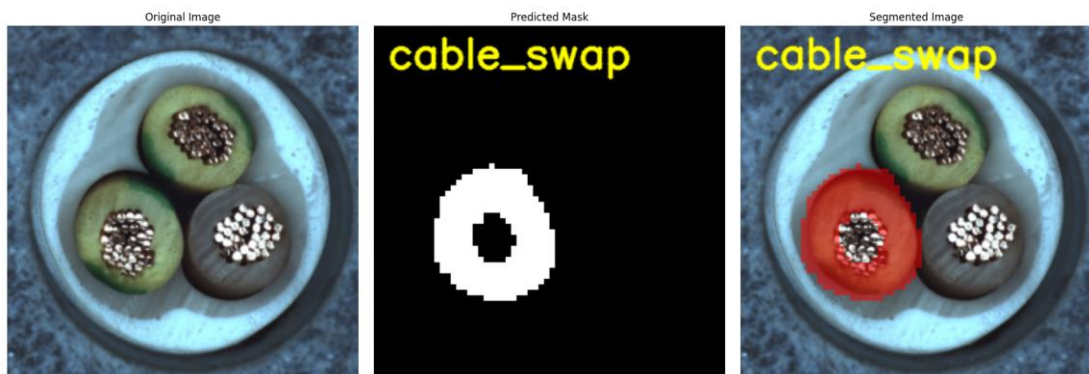


Figure 4. 35: Cable swap defect of cable



Figure 4. 36: Combined defect of cable

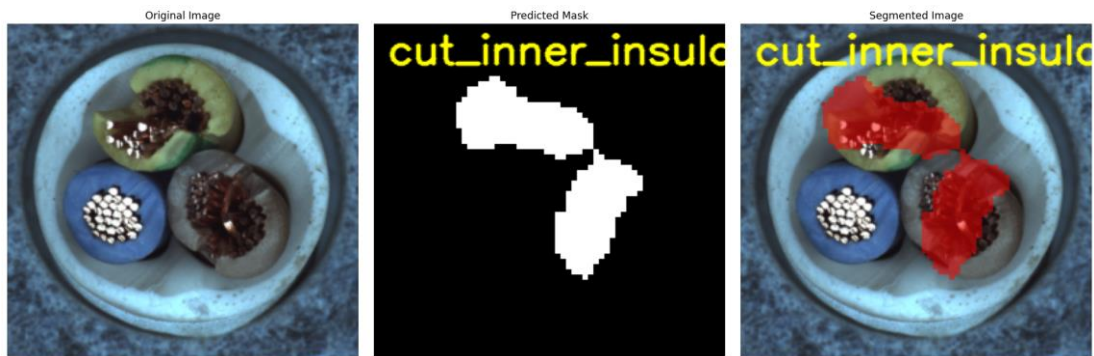


Figure 4. 37: Cut inner insulation defect of cable

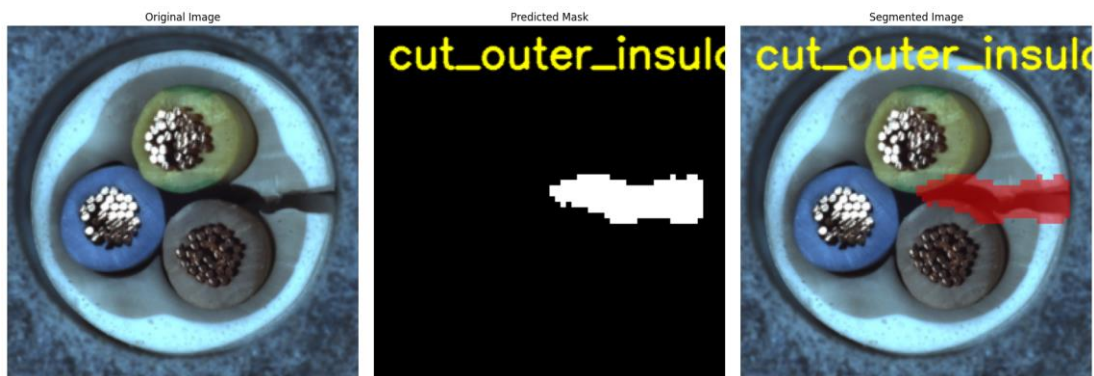


Figure 4. 38: Cut outer insulation defect of cable

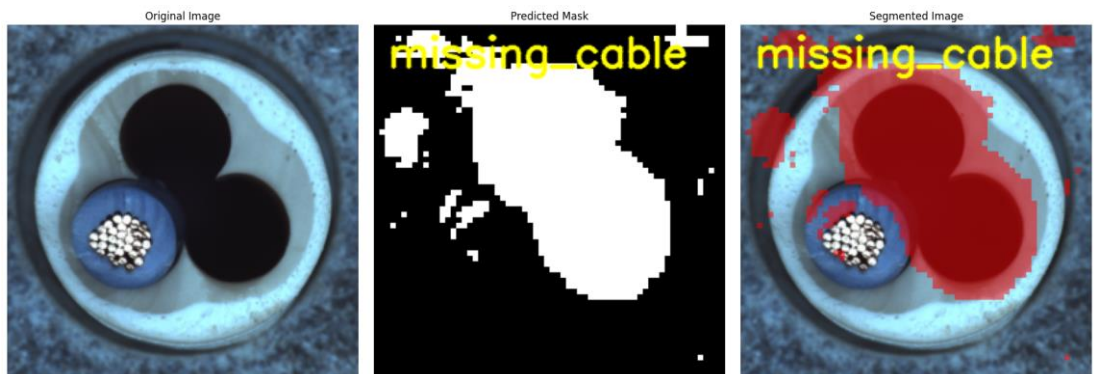


Figure 4. 39: Missing cable defect of cable

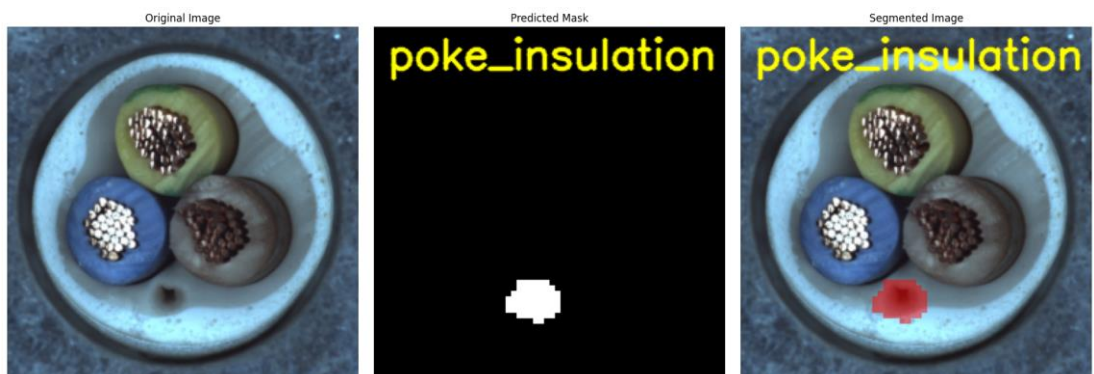


Figure 4. 40: Poke insulation defect of cable

4.3.2 Results obtained for Defect detection using segmentation for Grid using Custom U-Net model

This section describes learning curve from the trained model for grid, accuracy of the trained model for grid, predicted mask for the trained model for grid and Defected image with the predicted mask applied for grid.

4.3.2.1 Learning curve from the trained model for Grid using Custom U-Net model

The Figure 4.41(a) displays the training and validation loss and Figure 4.41(b) displays accuracy for a machine learning model over 35 epochs. In the loss graph, both training and validation loss decrease sharply in the initial epochs, stabilizing around epoch 5, indicating that the model is learning well without significant overfitting. The model achieves minimal loss around epoch 24, which is marked as the best epoch. The accuracy graph shows that both training and validation accuracy rise quickly, reaching a high and consistent performance near epoch 5. The best accuracy is observed at epoch 5, where the model reaches approximately 97.5% accuracy, suggesting early convergence. This behavior implies that the model generalizes well with minimal further improvement after early training stages, showcasing effective early learning.

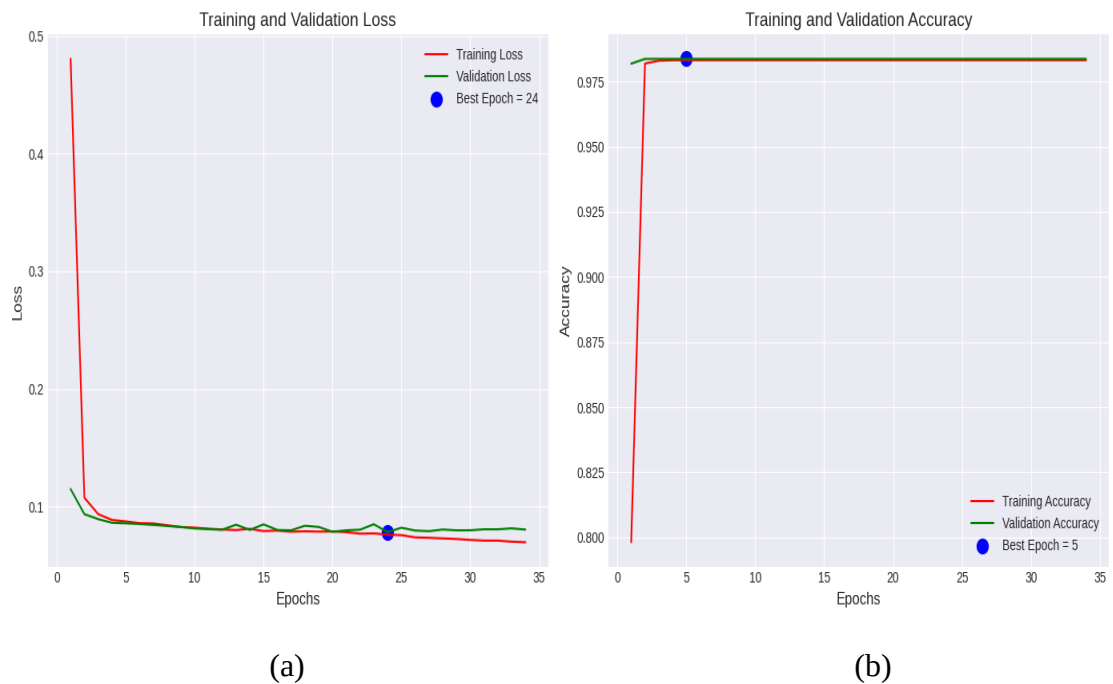


Figure 4. 41: Learning curve for Grid using Custom U-Net model

4.3.2.2 Accuracy of the trained model for Grid

This figure 4.42 shows the results of evaluating a machine learning model, likely a neural network (as indicated by the `model U-Net`), on a test dataset (`X`, `Y`). The key metrics here are:

- 1. Accuracy:** The model achieved an impressive accuracy of 0.9849 (or 98.49%), which means that nearly 98.5% of the predictions made by the model were correct.
- 2. Loss:** The loss value is 0.0716, which indicates a relatively low error rate. A lower loss generally means that the model's predictions are closely aligned with the actual targets.
- 3. Execution Speed:** The model evaluation ran at 85 ms/step, showing that it processed the data relatively quickly (especially given the number of steps in this evaluation: 2 in total).

These results suggest that the model performs very well, with high accuracy and low loss, likely indicating a well-optimized and well-generalized model. However, further testing on new data could confirm whether these metrics hold consistently across unseen data.

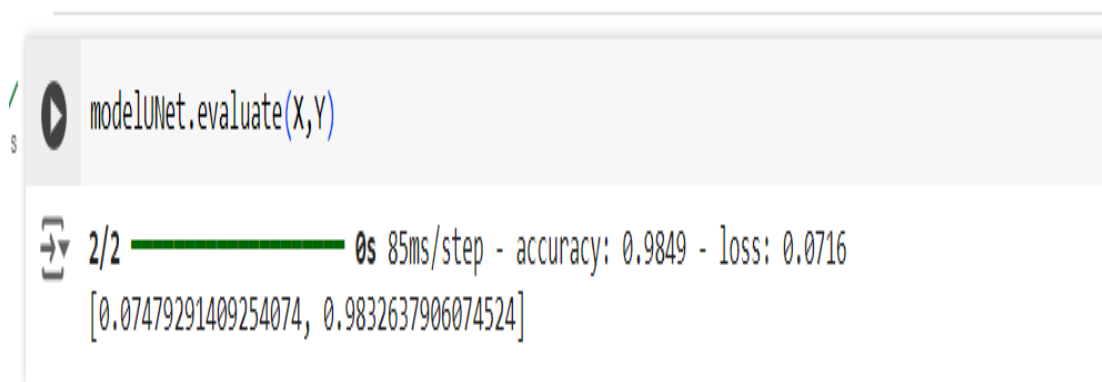


Figure 4. 42: Accuracy of the trained model for Grid

4.3.2.3 Predicted mask for the trained model for Grid

Figure 4.43 shows the predicted mask of the different defected area of the grid after training the model.

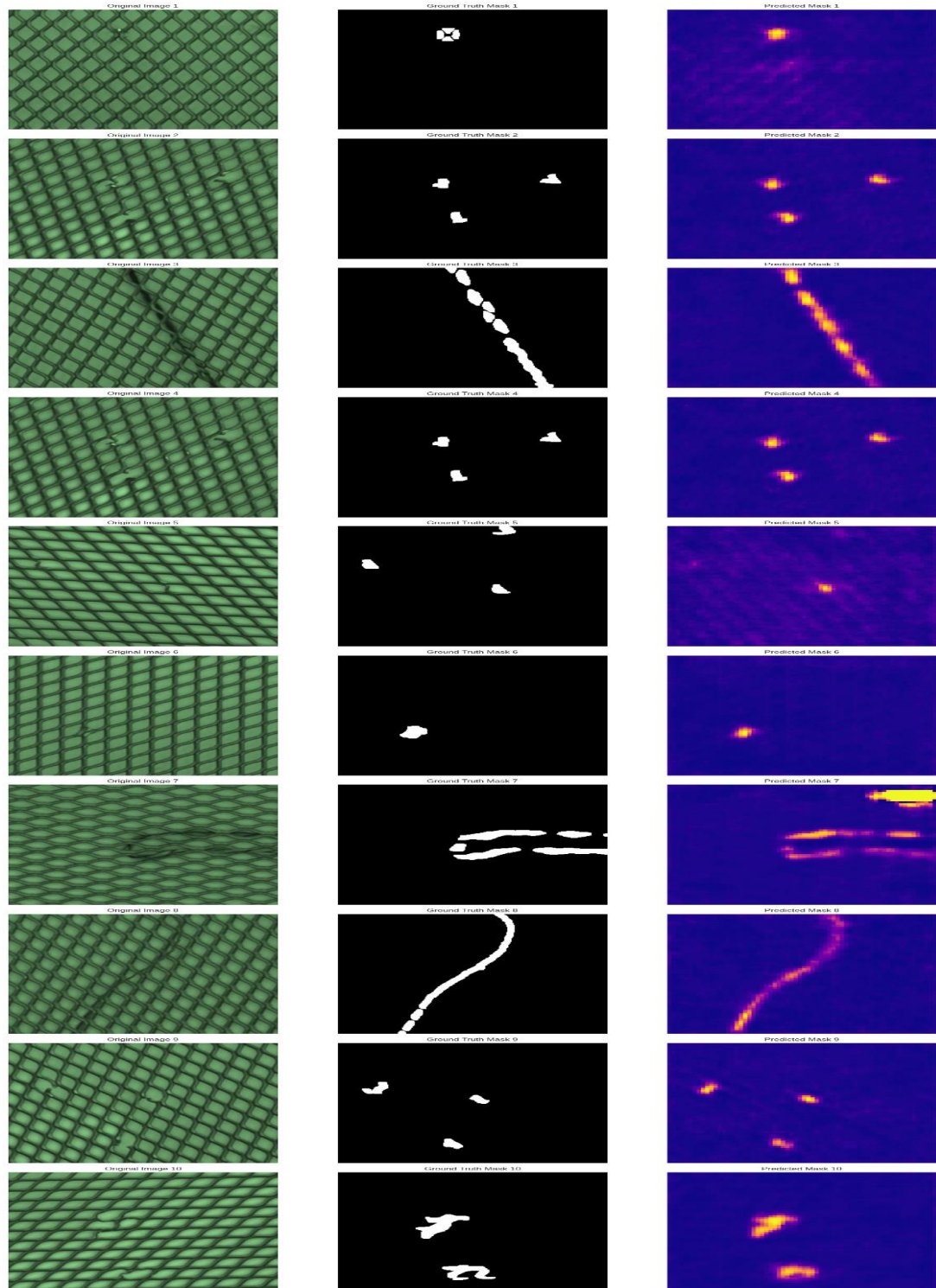


Figure 4. 43: Predicted mask for the trained model for Grid

4.3.2.4 Defected image with the predicted mask applied for Grid

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of grid like bent, broken, glue defect are segmented fault in Figure 4.44-Figure 4.46.

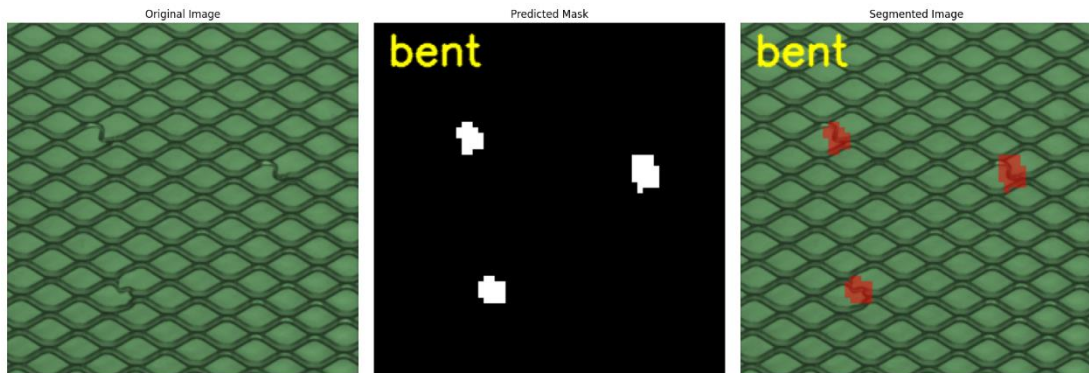


Figure 4. 44: Bent defect of Grid

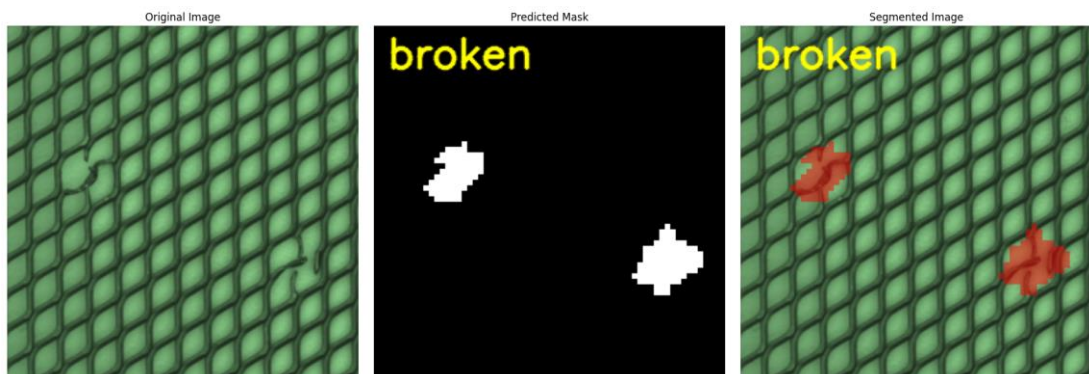


Figure 4. 45: Broken defect of Grid

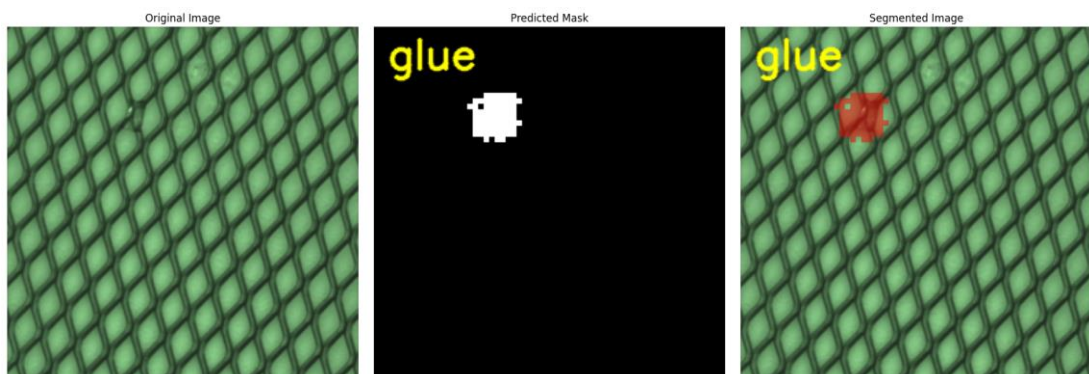


Figure 4. 46: Glue defect of grid

4.3.3 Results obtained for Defect detection using segmentation for Metal nut using Custom U-Net model

This section describes learning curve from the trained model for Metal nut, accuracy of the trained model for Metal nut, predicted mask for the trained model for Metal nut and Defected image with the predicted mask applied for Metal nut.

4.3.3.1 Learning curve from the trained model for Metal nut using Custom U-Net model

The Figure 4.47(a) illustrates the training and validation loss and Figure 4.47(b) illustrates accuracy (right) over 50 epochs for a machine learning model. The training loss starts extremely high but drops drastically within the first few epochs, stabilizing around epoch 5 and reaching a near-zero value by epoch 49, which is marked as the best epoch for the loss. The validation loss shows a similar rapid decrease and stabilizes early on, indicating good generalization with minimal overfitting. In contrast, the accuracy graph reveals a noticeable gap between training and validation accuracy. The validation accuracy quickly rises to about 94% by epoch 2, staying consistent throughout the remaining epochs, whereas the training accuracy remains lower, hovering around 82%, suggesting underfitting. This underfitting could imply that the model is not learning enough from the training data, even though it performs well on the validation set. Adjustments to model complexity or training strategies might be necessary to bridge this accuracy gap.

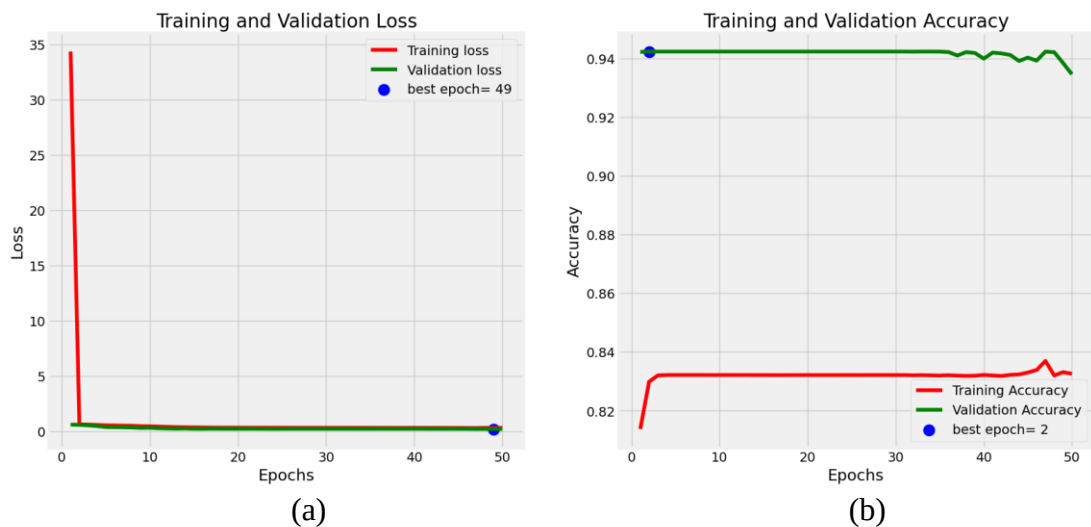


Figure 4. 47: Learning curve for Metal nut using Custom U-Net model

4.3.3.2 Accuracy of the trained model for Metal nut

This figure 4.48 represents the evaluation results of a machine learning model on a dataset ('X', 'Y'), providing insight into the model's performance:

1. Accuracy: The model achieved an accuracy of 0.8792 (or 87.92%), meaning that approximately 88% of the model's predictions were correct. While this is a decent accuracy, there is room for improvement depending on the use case or target performance level.

2. Loss: The loss value is 0.2821, which reflects the model's prediction error. Although not too high, this loss indicates that the model still has some discrepancies between its predictions and the actual target values. The loss might need to be reduced through further tuning, such as hyperparameter adjustment or additional training epochs.

3. Execution Speed: The evaluation took 16 seconds at a rate of 4 seconds per step, across 3 steps. This suggests that the model is moderately fast in terms of evaluation, though optimization might further enhance its speed.

In summary, the model demonstrates decent performance but could benefit from further improvements to reduce the loss and increase accuracy, especially if higher precision is required for the task.

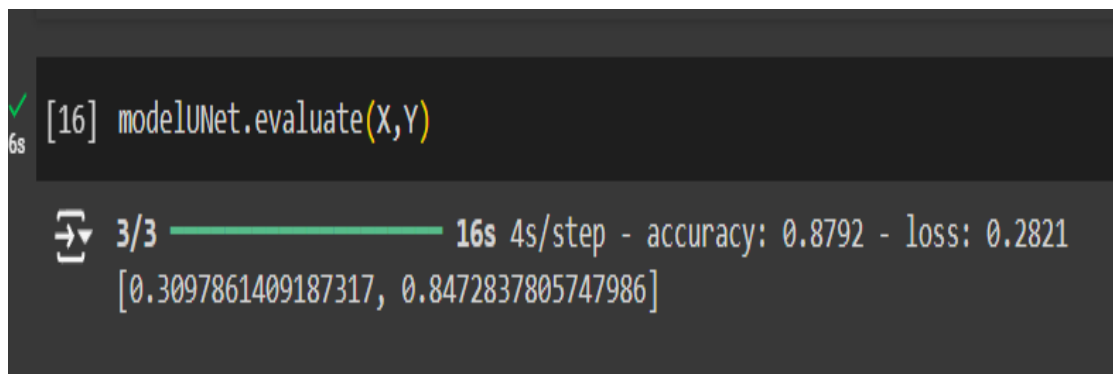


Figure 4. 48: Accuracy of the trained model for Metal nut

4.3.3.3 Predicted mask for the trained model for Metal nut

Figure 4.49 shows the predicted mask of the different defected area of the cable after training the model.

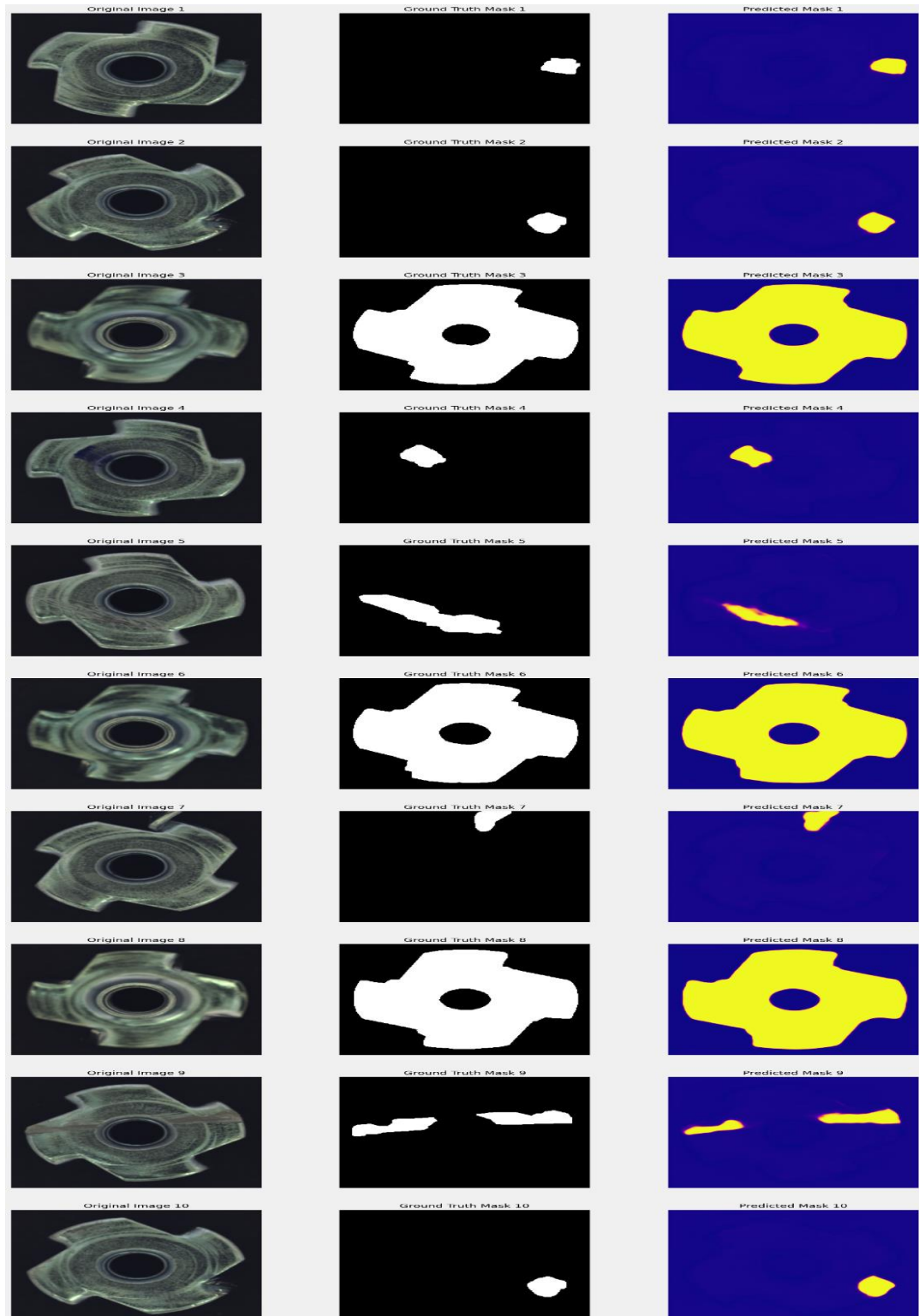


Figure 4. 49: Predicted mask for the trained model for Metal nut

4.3.3.4 Defected image with the predicted mask applied for Metal nut

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of Metal nut like bent, color, flip, scratch defect are segmented fault in Figure 4.50-Figure 4.53.

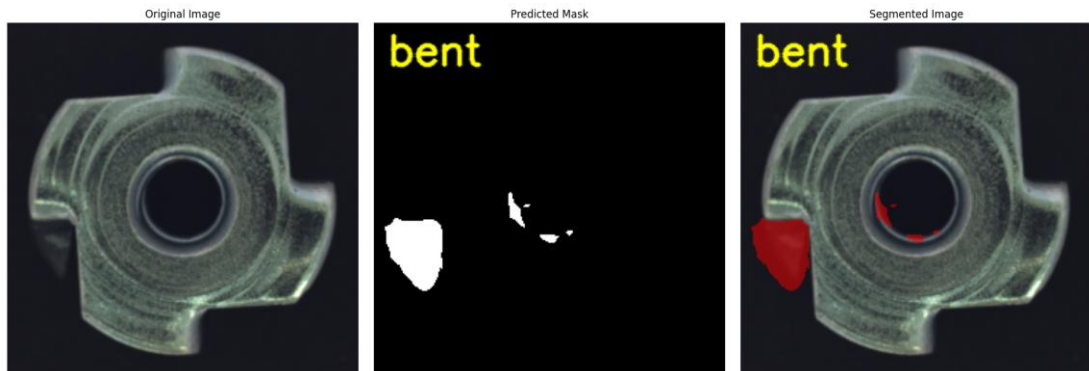


Figure 4. 50: Bent defect of metal nut

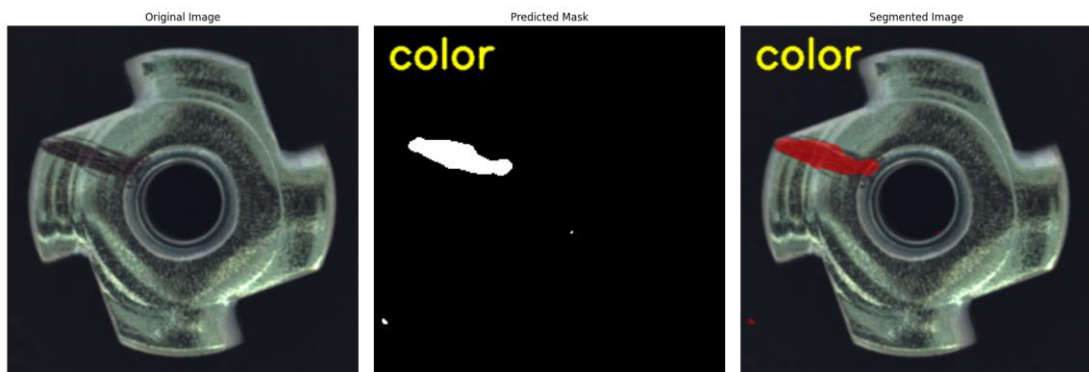


Figure 4. 51: Color defect of metal nut

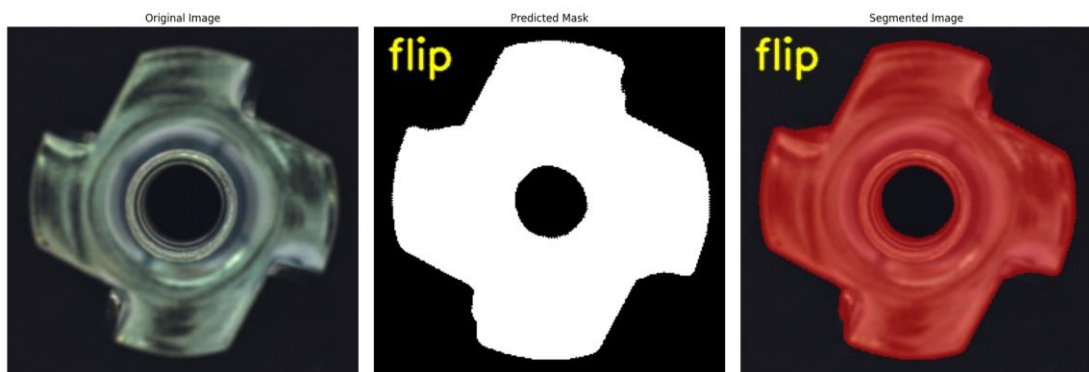


Figure 4. 52: Flip defect of metal nut

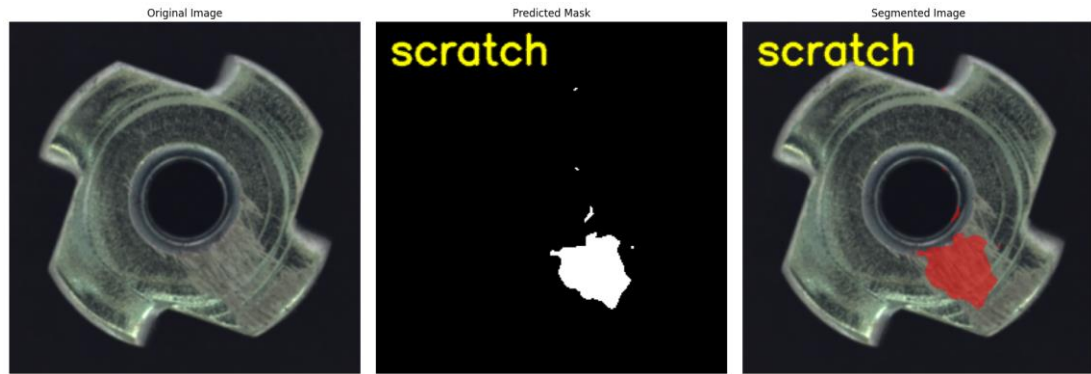


Figure 4. 53: Scratch defect of metal nut

4.3.4 Results obtained for Defect detection using segmentation for Screw using Custom U-Net model

This section describes learning curve from the trained model for screw, accuracy of the trained model for screw, predicted mask for the trained model for screw and Defected image with the predicted mask applied for screw.

4.3.4.1 Learning curve from the trained model for Screw using Custom U-Net model

The Figure 4.54(a) displays the training and validation loss and Figure 4.54(b) displays accuracy over 50 epochs for a machine learning model. The training and validation loss decrease rapidly during the first few epochs, both stabilizing around a low value after epoch 10, with the best epoch marked at 49. This suggests effective learning with minimal overfitting, as both losses converge to nearly identical, low values. On the accuracy side, both training and validation accuracy reach nearly 99% early, around epoch 5, and remain consistent throughout the training process. The model demonstrates excellent performance in terms of both learning from the training data and generalizing to unseen data, achieving almost perfect accuracy. The lack of significant divergence between the training and validation curves indicates the model is neither overfitting nor underfitting, and it has reached optimal performance.

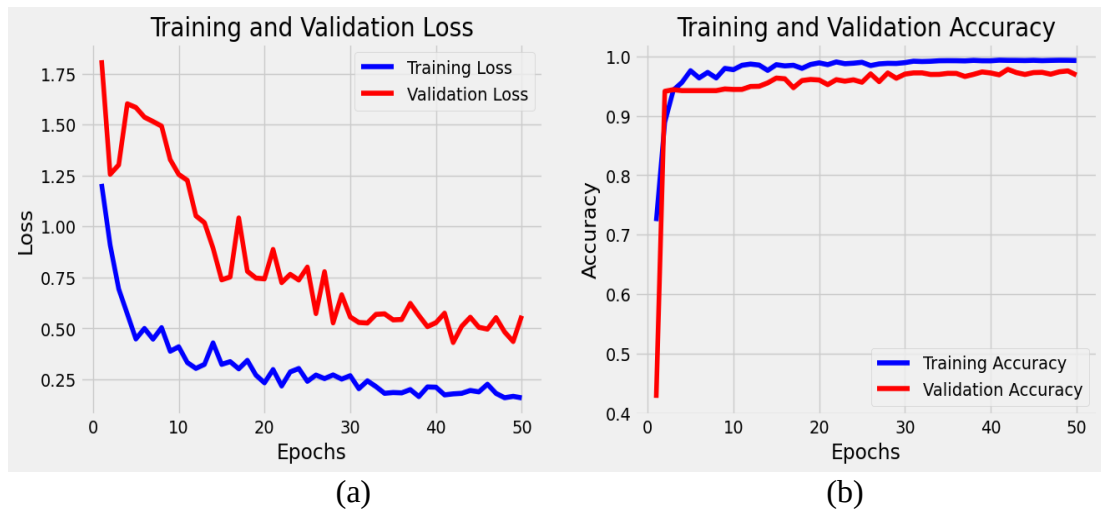


Figure 4. 54: Learning curve for screw using Custom U-Net model

4.3.4.2 Accuracy of the trained model for Screw

This figure 4.55 shows the evaluation of a neural network model (`model U-Net`) on a dataset (`X`, `Y`). The key performance metrics are:

- 1. Accuracy:** The model achieved an exceptionally high accuracy of 0.9958 (or 99.58%), meaning that nearly all predictions made by the model were correct. This suggests that the model has been very well trained and generalized to the evaluation dataset.
- 2. Loss:** The loss is extremely low at 0.0147, which reflects minimal prediction errors. A loss this small indicates that the model's outputs closely match the actual target values, further reinforcing the high performance.
- 3. Execution Speed:** The evaluation was fast, processing the data at 92 ms/step over 4 steps, indicating that the model is highly efficient in terms of computational performance.

Overall, these results demonstrate an exceptionally well-performing model with both high accuracy and low error rates, making it highly reliable for the task at hand.

```
# Evaluate the model

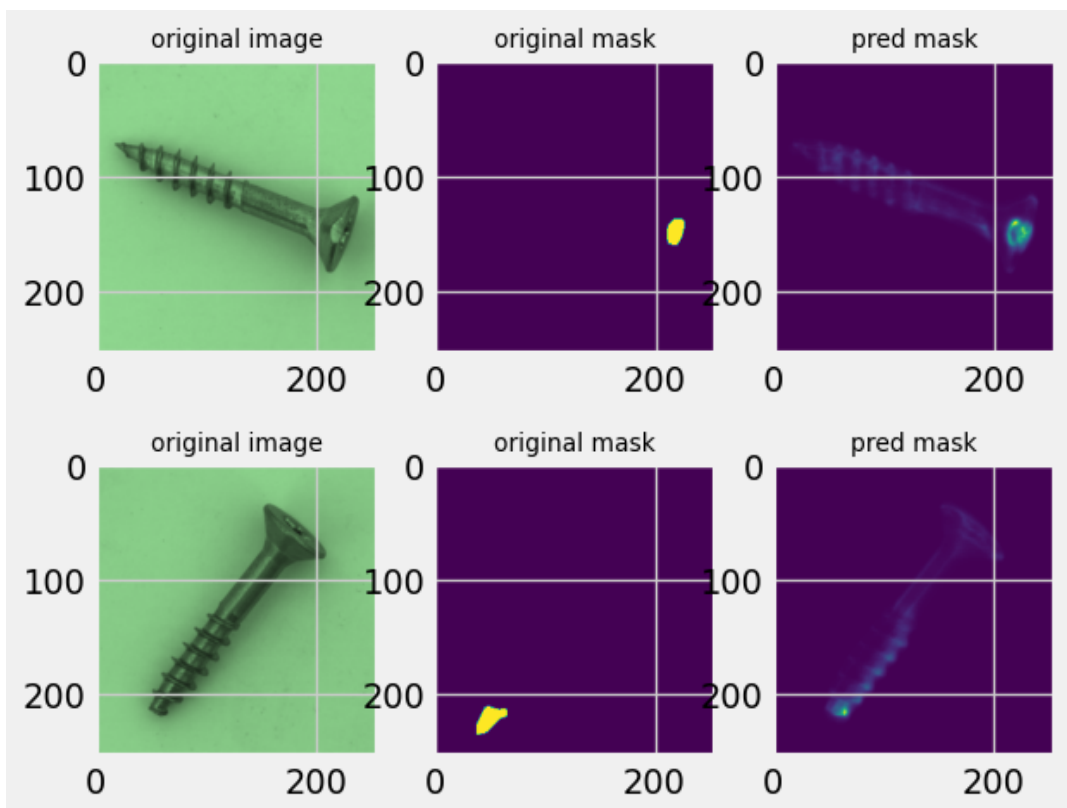
modelUNet.evaluate(X, Y)
```

4/4 ————— 0s 92ms/step - accuracy: 0.9958 - loss: 0.0147
[0.01362259779125452, 0.9959353804588318]

Figure 4. 55: Accuracy of the trained model for Screw

4.3.4.3 Predicted mask for the trained model for Screw

Figure 4.56 shows the predicted mask of the different defected area of the cable after training the model.



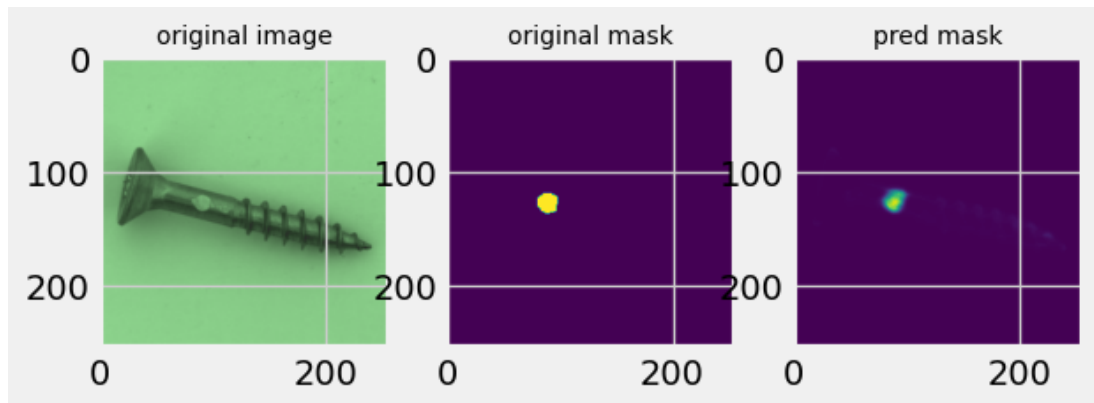


Figure 4. 56: Predicted mask for the trained model for Screw

4.3.4.4 Defected image with the predicted mask applied for Screw

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of screw like manipulated front, scratch head, scratch neck, thread side, thread top defect are segmented fault in Figure 4.57-Figure 4.61.



Figure 4. 57: Manipulated front defect of screw

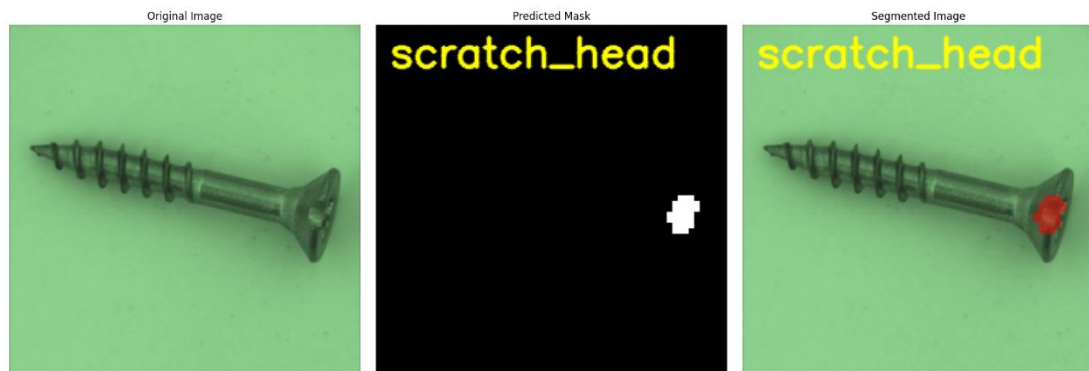


Figure 4. 58: Scratch head defect of screw

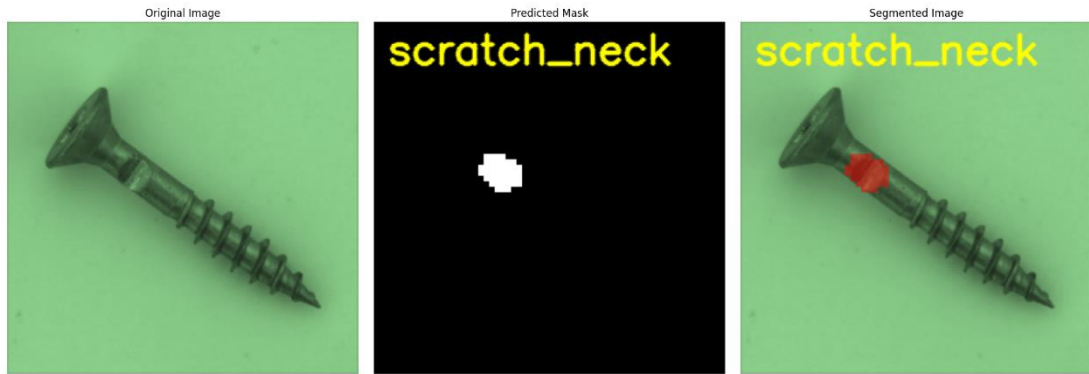


Figure 4. 59: Scratch neck defect of screw

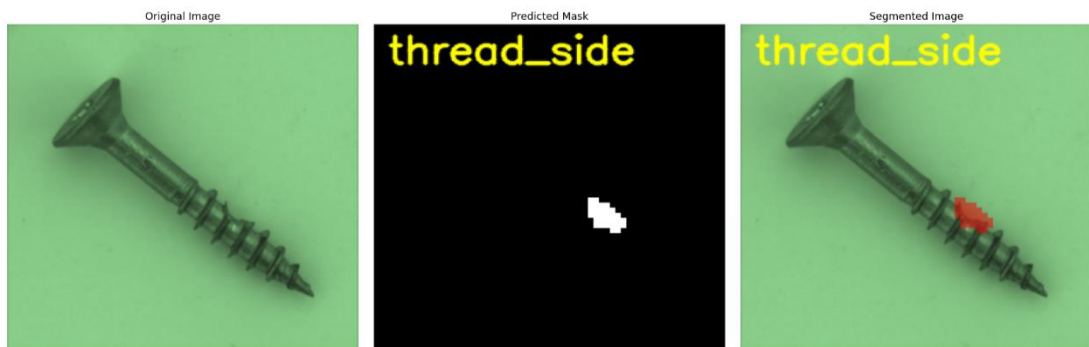


Figure 4. 60: Thread side defect of screw

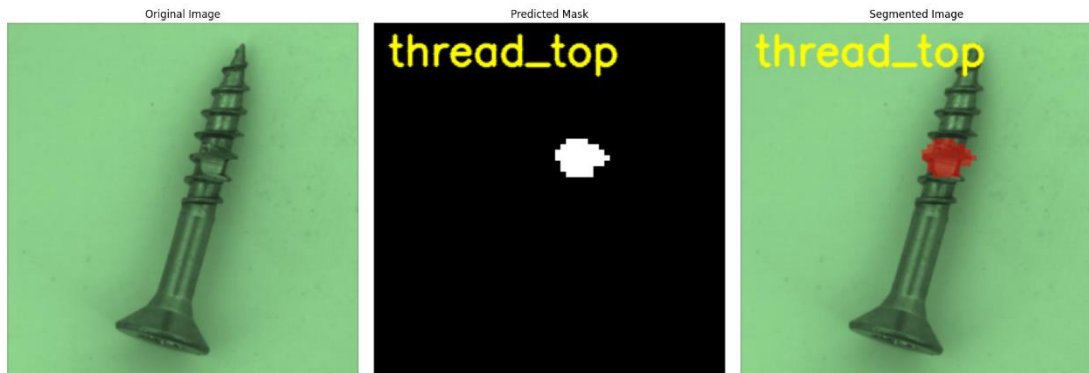


Figure 4. 61: Thread top defect of screw

4.3.5 Results obtained for Defect detection using segmentation for Transistor using Custom U-Net model

This section describes learning curve from the trained model for transistor, accuracy of the trained model for transistor, predicted mask for the trained model for transistor and Defected image with the predicted mask applied for transistor.

4.3.5.1 Learning curve from the trained model for Transistor using Custom U-Net model

The Figure 4. 62(a) depicts the training and validation loss and Figure 4.62(b) depicts accuracy over 50 epochs. In the loss graph, the training loss decreases steadily, with noticeable fluctuations throughout the epochs, suggesting that the model has not fully stabilized during training. The validation loss is consistently lower than the training loss, reaching its minimum at epoch 50, marked as the best epoch. The validation loss curve also exhibits fluctuations, though more stabilized compared to the training loss. On the accuracy side, the validation accuracy reaches a perfect 1.0 (100%) as early as epoch 2 and remains flat throughout the training process, while the training accuracy plateaus around 87% after an initial sharp rise. This significant gap between training and validation accuracy suggests that the model may be overfitting, as it generalizes extremely well on the validation data but struggles to improve training performance. Adjustments such as more regularization or changes to model complexity may be needed to address this discrepancy.

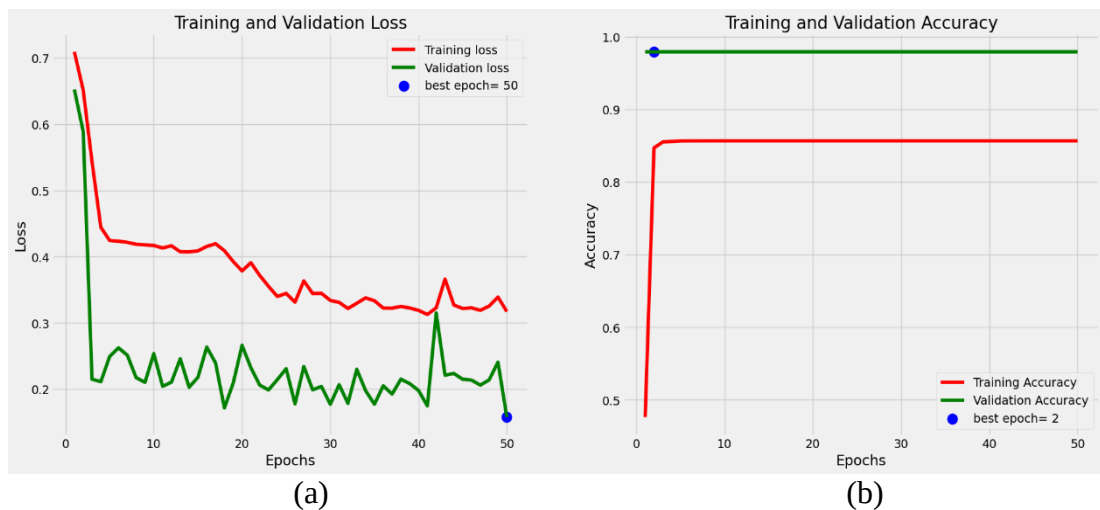


Figure 4. 62: Learning curve for transistor using Custom U-Net model

4.3.5.2 Accuracy of the trained model for Transistor

This figure 4.63 shows the evaluation of a neural network model (`model U-Net`) on a dataset (`X`, `Y`). The key metrics provided are:

1. Accuracy: The model achieved an accuracy of 0.9879 (or 98.79%), meaning that about 98.8% of the model's predictions were correct. This reflects strong performance, indicating that the model can reliably predict outcomes with high precision.

2. Loss: The loss value is 0.0297, which is relatively low, indicating that the model's prediction errors are small and that it closely matches the expected target values.

3. Execution Speed: The evaluation processed data at 245 ms/step over 4 steps. While slightly slower than some of the previous evaluations, this speed is still efficient for most tasks.

Overall, the model demonstrates very good performance with high accuracy and low loss, indicating that it has been well-trained and generalizes effectively to the evaluation data. The slight decrease in speed may be due to the model's complexity or the evaluation environment, but it remains within acceptable limits.

```
# Evaluate the model

modelUNet.evaluate(X, Y)
```

4/4 ————— 1s 245ms/step - accuracy: 0.9879 - loss: 0.0297
[0.03005167841911316, 0.9880924820899963]

Figure 4. 63: Accuracy of the trained model for Transistor

4.3.5.3 Predicted mask for the trained model for Transistor

Figure 4.64 shows the predicted mask of the different defected area of the cable after training the model

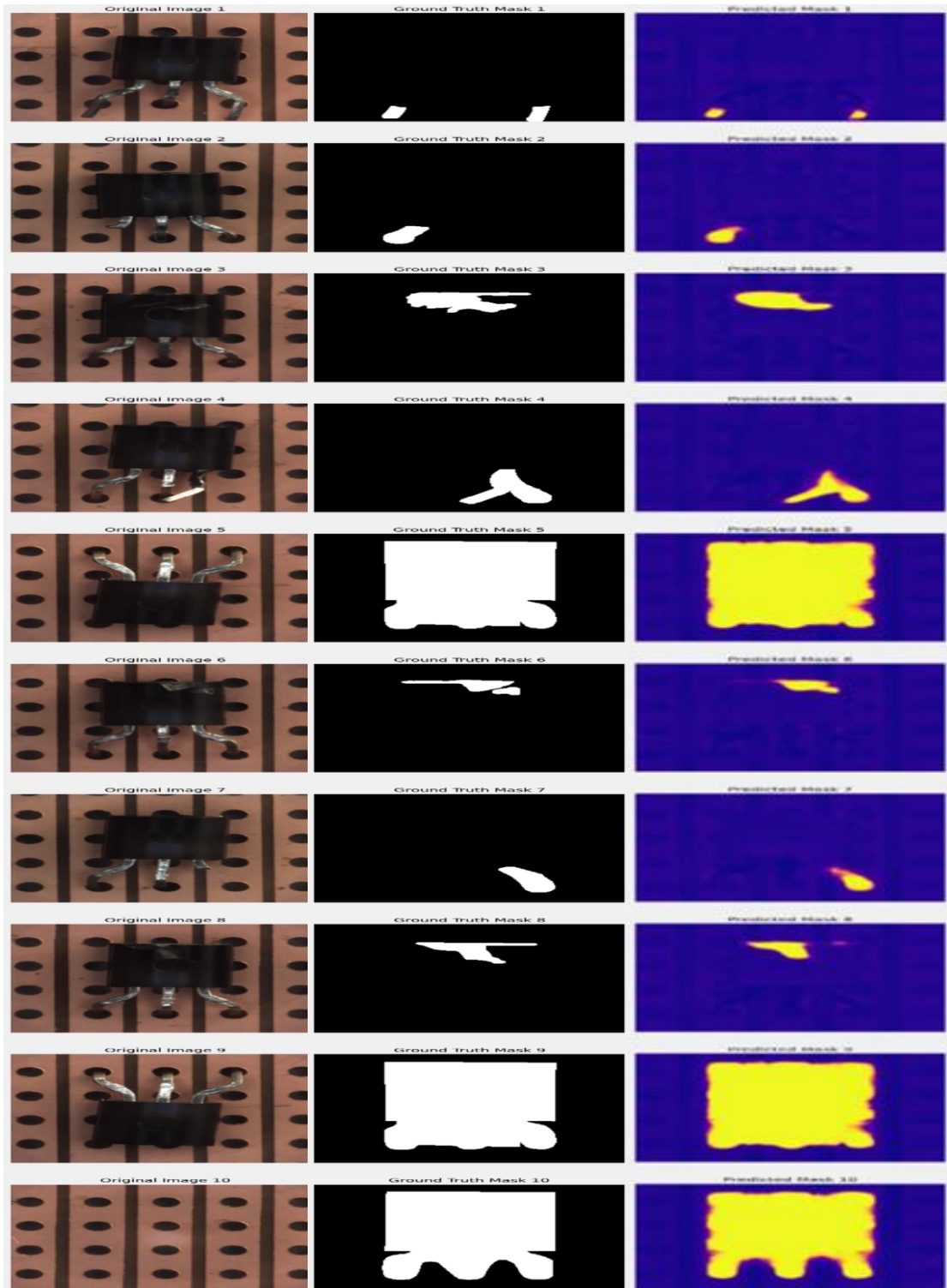


Figure 4. 64: Predicted mask for the trained model for Transistor

4.3.5.4 Defected image with the predicted mask applied for Transistor

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of transistor like bent lead, cut lead, misplaced, damaged case defect are segmented fault in Figure 4.65-Figure 4.68.

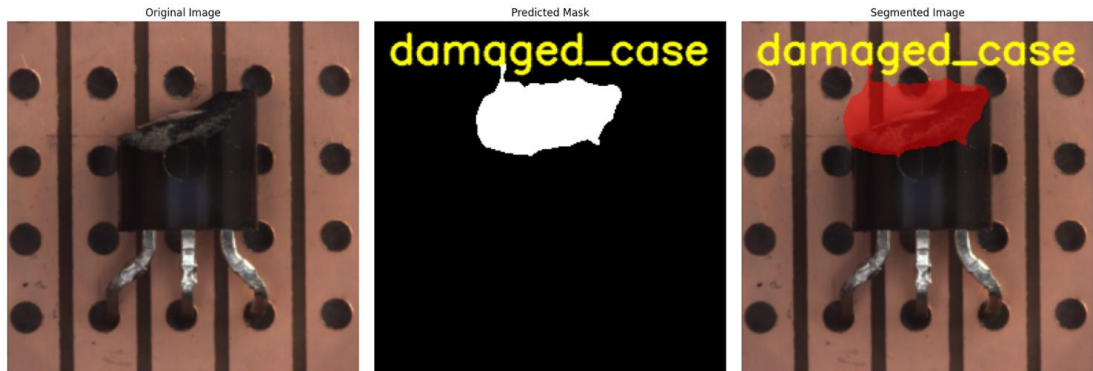


Figure 4. 65: Damaged case defect of transistor

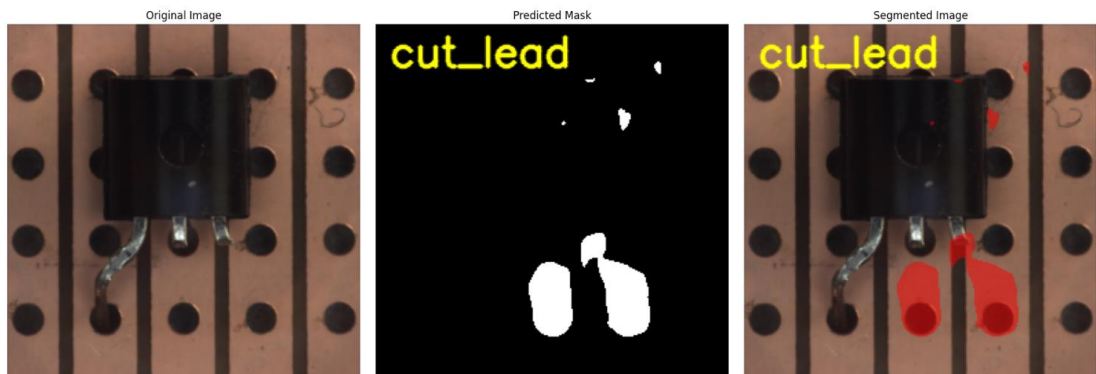


Figure 4. 66: Cut lead defect of transistor

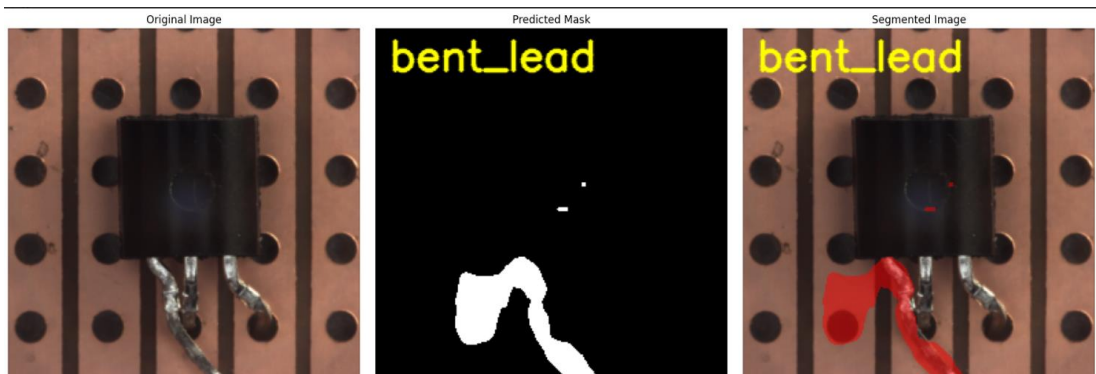


Figure 4.67: Bent lead defect of transistor

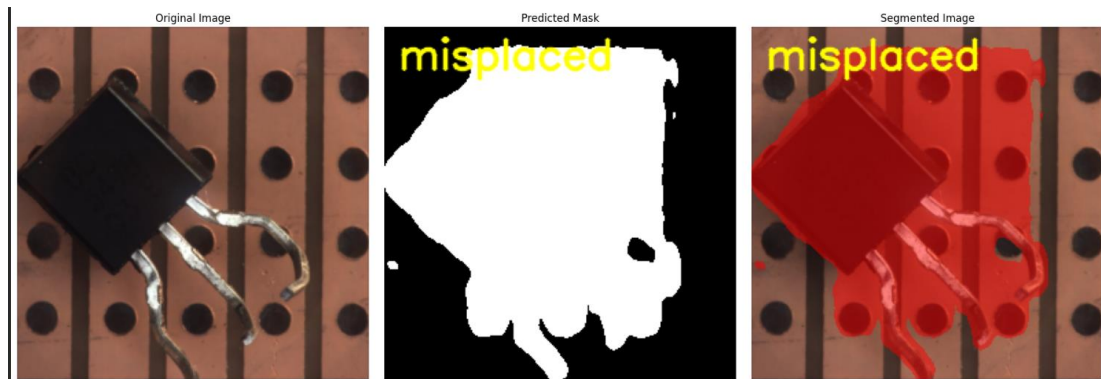


Figure 4. 68: Misplaced defect of transistor

4.3.6 Results obtained for Defect detection using segmentation for Zipper using Custom U-Net model

This section describes learning curve from the trained model for Zipper, accuracy of the trained model for Zipper, predicted mask for the trained model for Zipper and Defected image with the predicted mask applied for Zipper.

4.3.6.1 Learning curve from the trained model for Zipper using Custom U-Net model

The Figure 4.69(a) illustrates the model's training and validation performance over 50 epochs. The plot on the left presents the training and validation loss, where both losses decrease significantly during the initial 10 epochs, indicating rapid improvement in model performance. Beyond this point, the loss values stabilize near zero, with minimal divergence between the two curves. This suggests that the model is effectively learning from the data without overfitting. The optimal performance, marked by the blue dot, occurs at epoch 46, where the validation loss reaches its lowest point.

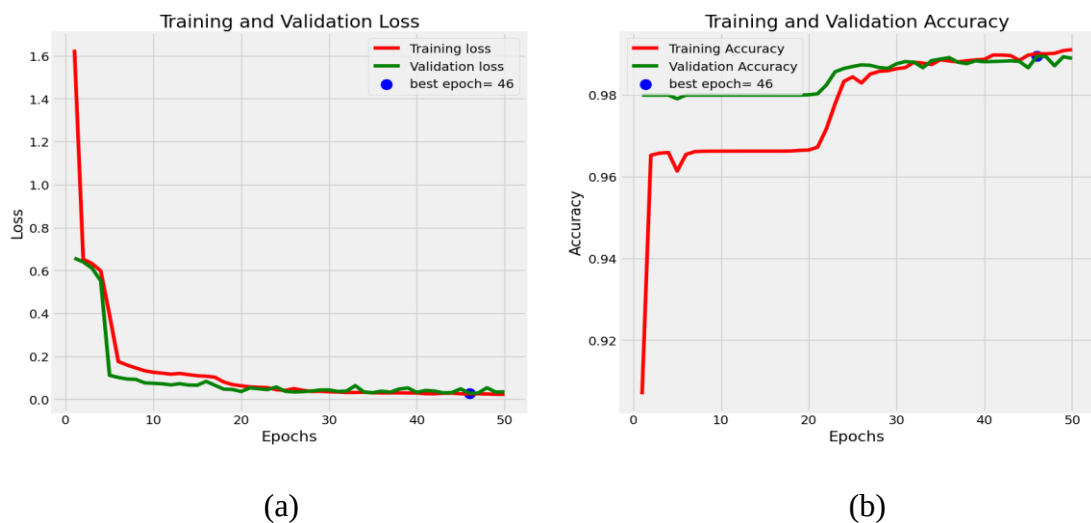


Figure 4. 69: Learning curve for Zipper using Custom U-Net model

The Figure 4.69(b) displays the training and validation accuracy. Both metrics show a sharp increase within the first few epochs, followed by a gradual convergence. The training accuracy initially lags behind the validation accuracy but quickly aligns after approximately 10 epochs. Both metrics stabilize above 98%, reflecting the model's high accuracy. The close alignment of the training and validation curves in both plots indicates strong generalization and suggests the model is not overfitting, as its performance on unseen validation data is consistent with the training data. This analysis demonstrates that the model is well-optimized, with minimal overfitting and excellent generalization across both loss and accuracy metrics.

4.3.6.2 Accuracy of the trained model for Zipper

This figure 4.70 represents the evaluation results of a neural network model (`model U-Net`) on a dataset (`X`, `Y`). Here are the key metrics:

- 1. Accuracy:** The model achieved an accuracy of 0.9879 (or 98.79%), indicating that nearly 99% of the predictions made by the model are correct. This suggests that the model is highly effective in its predictions.
- 2. Loss:** The loss value is 0.0297, which indicates a very low error rate between the model's predictions and the actual target values. A lower loss value is typically an indicator of better performance, meaning the model is well-calibrated.
- 3. Execution Speed:** The evaluation processed at 245 ms/step across 4 steps, which is a relatively moderate speed, likely influenced by the complexity of the model or dataset size.

Overall, this evaluation shows that the model is performing exceptionally well, with high accuracy and minimal loss, demonstrating its ability to generalize effectively on the evaluation dataset.

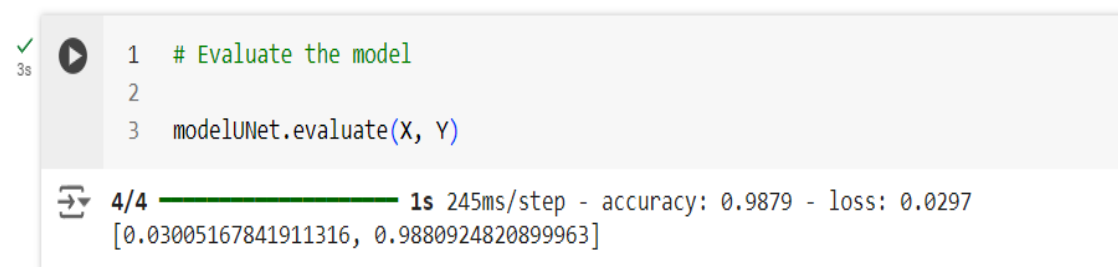


Figure 4.70: Accuracy of the trained model for Zipper

4.3.6.3 Predicted mask for the trained model for Zipper

Figure 4.71 shows the predicted mask of the different defected area of the cable after training the model

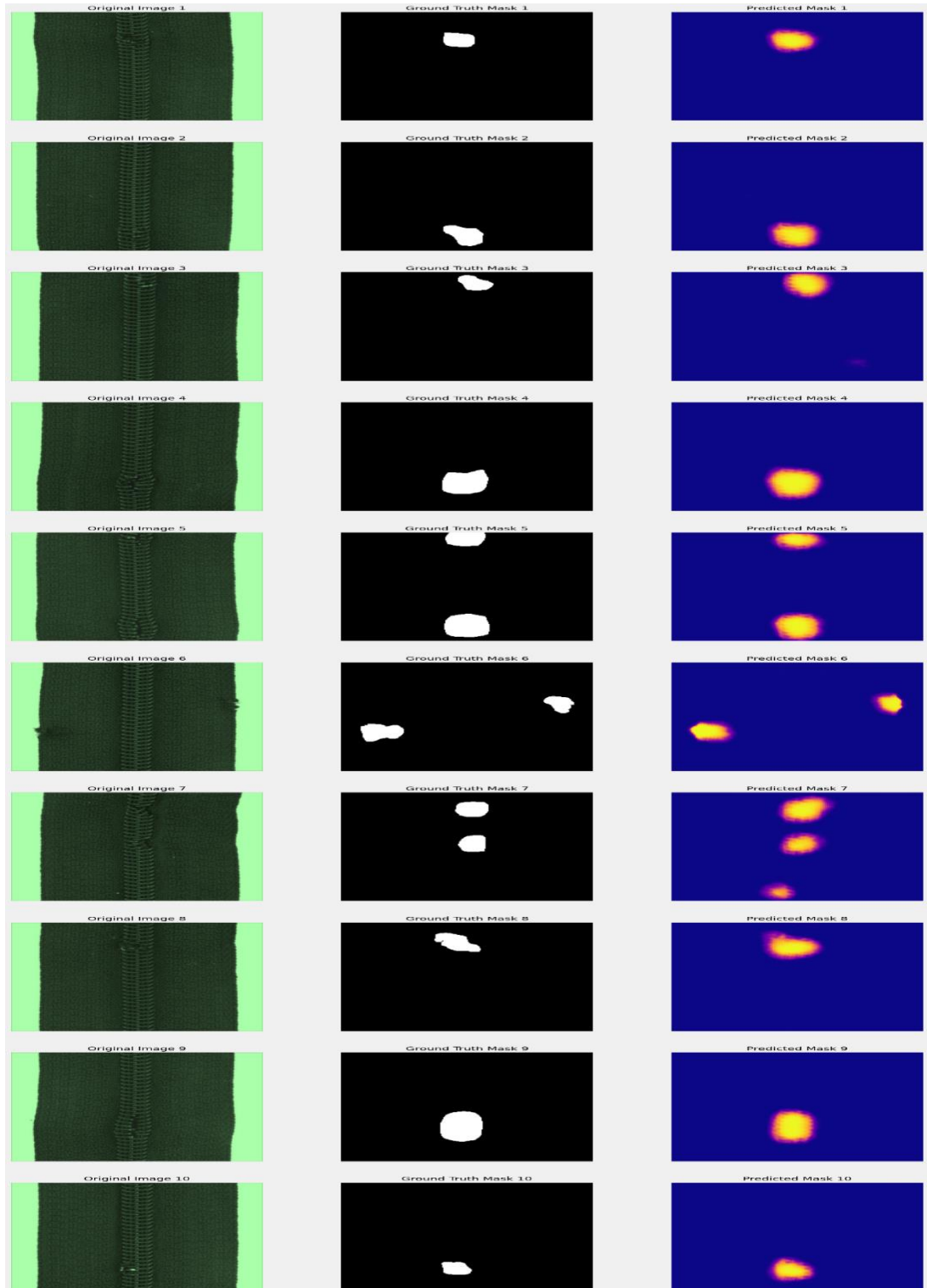


Figure 4.71: Predicted mask for the trained model for Zipper

4.3.6.4 Defected image with the predicted mask applied for Zipper

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of Zipper like fabric border defect, fabric interior defect, rough defect, squeezed teeth defect, split teeth defect are segmented fault in Figure 4.72-Figure 4.76.

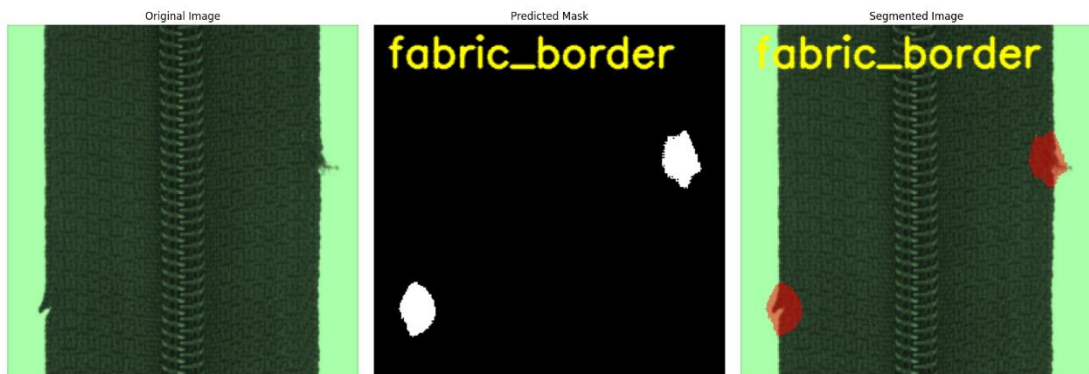


Figure 4. 72: Fabric border defect of zipper

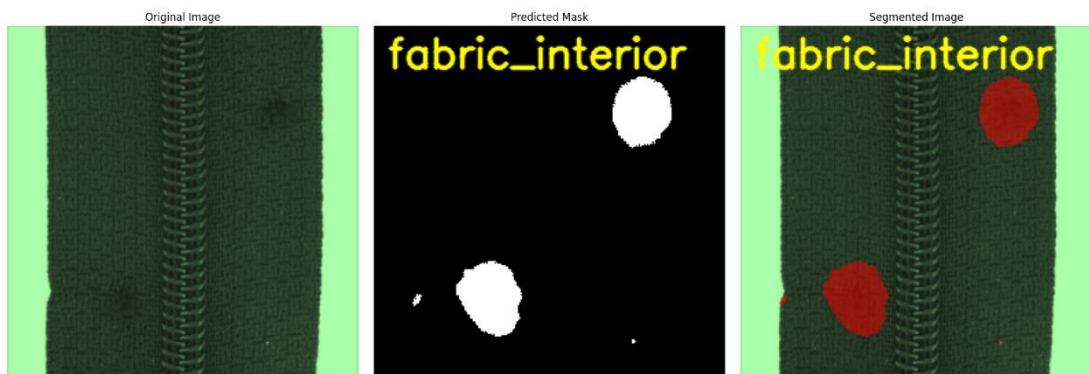


Figure 4.73: Fabric border interior of zipper

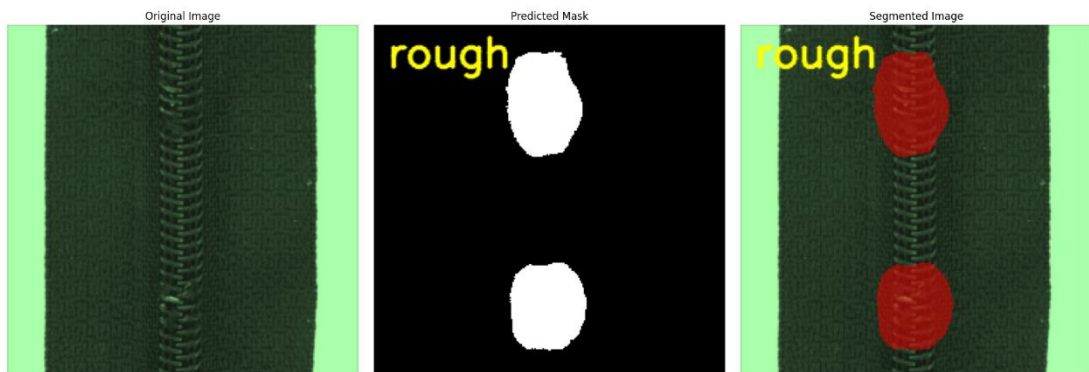


Figure 4.74: Rough defect of zipper



Figure 4. 75: Split teeth defect of zipper

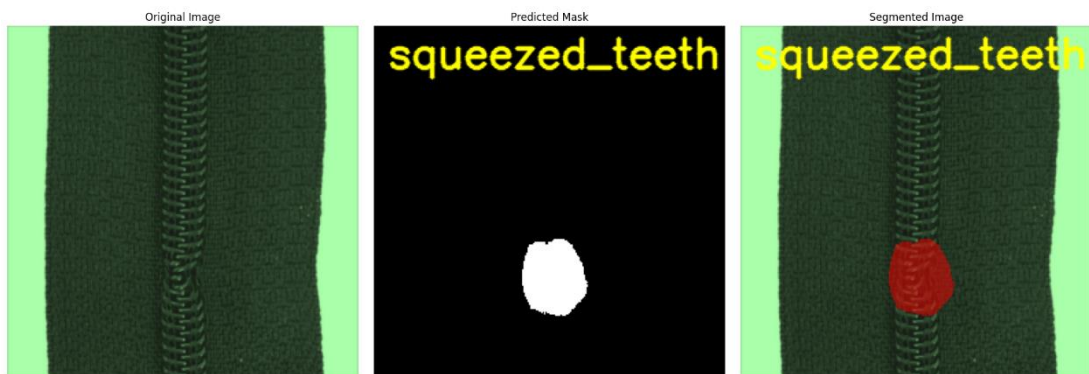


Figure 4.76: Squeezed teeth defect of zipper

The Table 4.2 illustrates the performance of a Custom U-Net model across different object types in terms of accuracy. This level of accuracy suggests that the model is robust for applications requiring high precision in identifying these specific object types.

Table 4. 2: Accuracy for different objects using Custom U-Net model

	Object Type					
Algorithm	Cable	Grid	Metal nut	Screw	Transistor	Zipper
Custom U-Net model	96.92%	98.33%	87.92%	99.59%	98.79%	98.79%

4.4 Results obtained for Defect detection using segmentation on metallic products using DeeplabV3+ model

Defect detection on metallic products is a crucial application in industries such as manufacturing, where quality control plays a vital role in ensuring product reliability.

Deep learning techniques, particularly DeepLabV3+ segmentation models have become effective tools for automating this process, enabling the identification of surface defects with high accuracy and efficiency. DeeplabV3+ is an excellent choice for Defect detection on metallic products due to its ability to provide precise, pixel-level segmentation. By leveraging this model, industries can automate and improve their quality control processes, ensuring that defects are detected early and accurately. With advancements in hardware and optimization techniques, DeeplabV3+ can also be adapted for real-time applications, making it an invaluable tool in modern industrial automation.

4.4.1 Results obtained for Defect detection and segmentation for cable using DeeplabV3+ model

This section describes learning curve from the trained model for cable, accuracy of the trained model for cable, predicted mask for the trained model for cable and Defected image with the predicted mask applied for Cable.

4.4.1.1 Learning curve from the trained model for cable using DeeplabV3+ model

The learning curves presented in the graphs provide a comprehensive view of the model's training and validation performance across 35 epochs. The Figure 4. 77(a) illustrates the training and validation loss, while the Figure 4. 77(b) depicts the training and validation accuracy. These curves offer valuable insights into the model's convergence and its ability to generalize to unseen data.

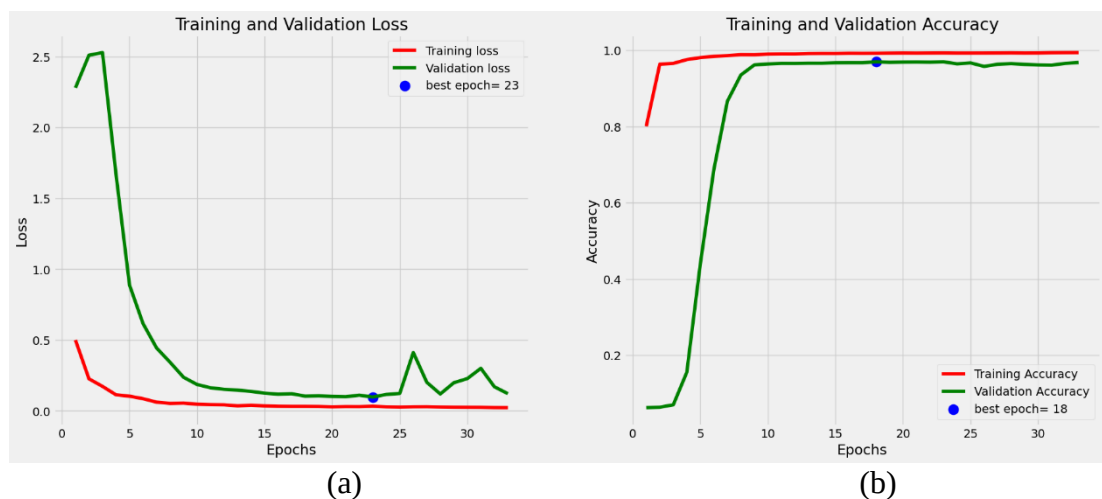


Figure 4. 77: Learning curve for cable using DeeplabV3+ model

The loss curves indicate a significant decline in both training and validation loss during the initial epochs, reflecting the model's ability to learn rapidly from the data. The training loss continues to decrease and reaches a minimal value close to zero after approximately 15 epochs, indicating that the model has achieved a high level of fit on the training data. However, the validation loss, after an initial sharp decrease, stabilizes around 0.2 and begins to exhibit slight fluctuations after epoch 23. This behavior suggests that while the model continues to improve on the training data, its performance on the validation data does not improve as consistently beyond a certain point, possibly indicating the onset of overfitting.

On the accuracy curves, the model's training accuracy shows a sharp increase and quickly approaches 100%, stabilizing after around 15 epochs. The validation accuracy, on the other hand, increases rapidly and levels off at approximately 98%, achieving optimal performance by epoch 18. The blue markers on both graphs indicate the "best" epochs: epoch 23 for validation loss and epoch 18 for validation accuracy, suggesting that the model generalizes well up to this point.

The slight divergence between the training and validation loss curves after epoch 23, combined with the stabilization of validation accuracy, may indicate the model's tendency to overfit the training data. Despite this, the model demonstrates strong generalization performance, as indicated by the validation accuracy curve. Further fine-tuning, such as regularization techniques or early stopping, could be explored to mitigate overfitting and optimize the model's performance.

These learning curves provide a clear indication of the model's training dynamics and its potential to generalize, serving as an important foundation for evaluating model performance in real-world applications.

4.4.1.2 Accuracy of the trained model for cable

The Figure 4.78 represents the evaluation of the DeepLabV3+ model for Defect detection on cable resulted in a high accuracy of 98.52% and a low loss of 0.0542,

indicating strong performance in segmenting defects at the pixel level. The model was tested across three batches, with each step taking around 9 seconds. The high accuracy reflects the model's capability to correctly identify defective regions on metallic products, while the low loss suggests minimal prediction error, highlighting the model's effectiveness in handling complex textures and variations commonly found on metallic surfaces. These results make the DeepLabV3+ model highly suitable for real-time industrial applications in quality control and Defect detection.

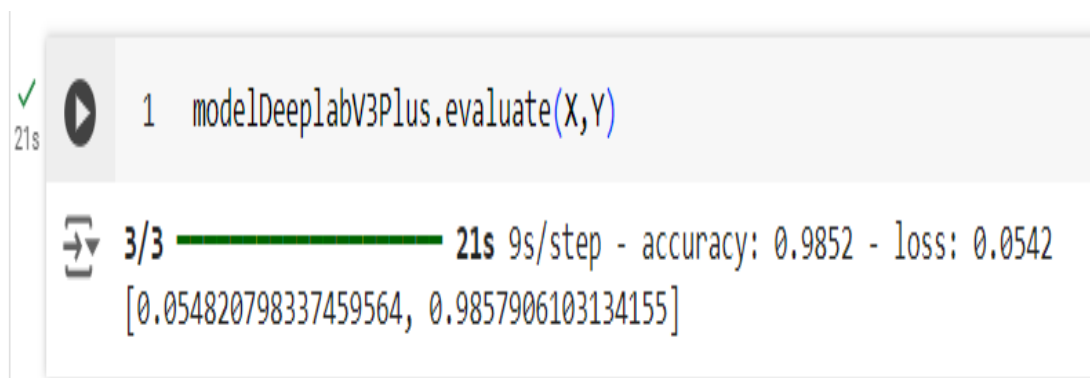


Figure 4.78: Accuracy of the trained model for cable

4.4.1.3 Predicted mask for the trained model for cable

Figure 4.79 shows the predicted mask of the different defected area of the cable after training the model.

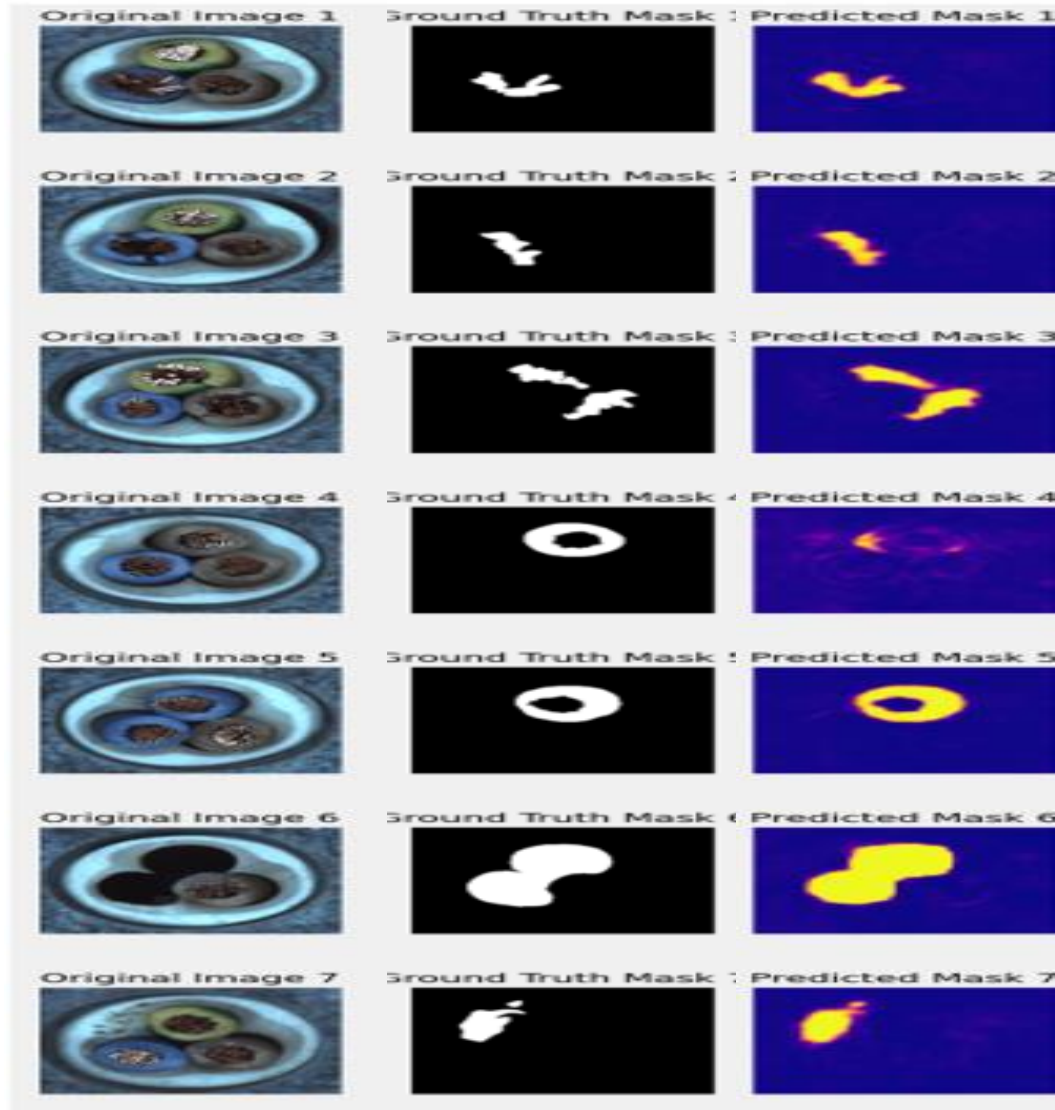


Figure 4.79: Predicted mask for the trained model for cable

4.4.1.4 Defected image with the predicted mask applied for cable

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of Cable like bent wire, cable swap, combined defect, cut inner insulation, cut outer insulation, missing cable and poke insulation defect are segmented fault in Figure 4.80-Figure 4.85.

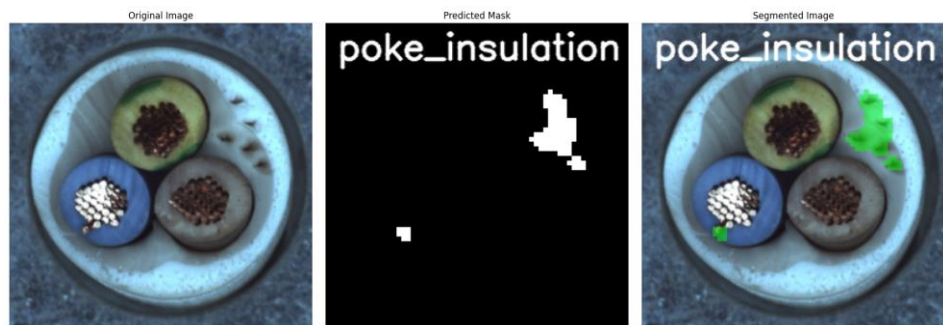


Figure 4.80: Poke insulation defect of Cable



Figure 4.81: Cut inner insulation defect of Cable

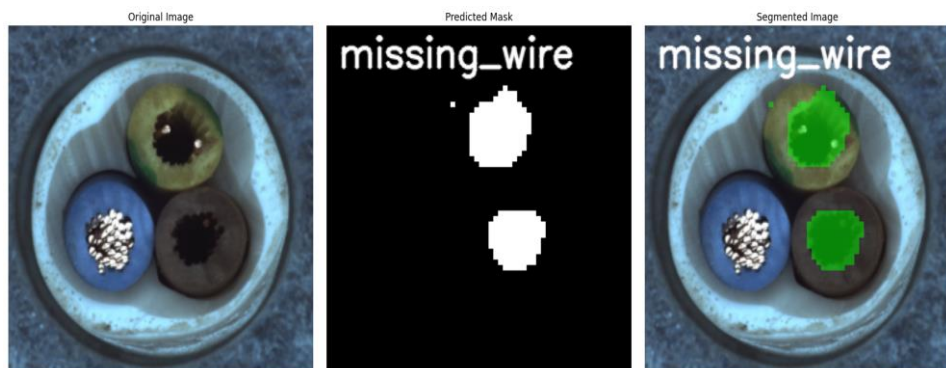


Figure 4.82: Missing wire defect of Cable

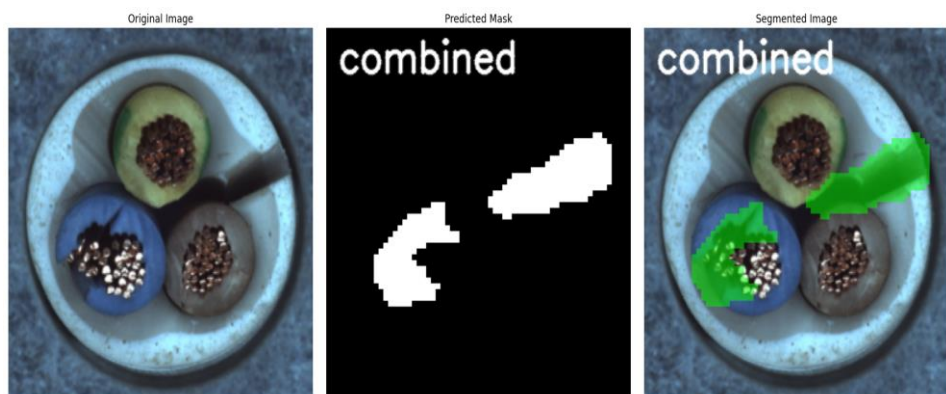


Figure 4.83: Combined defect of Cable

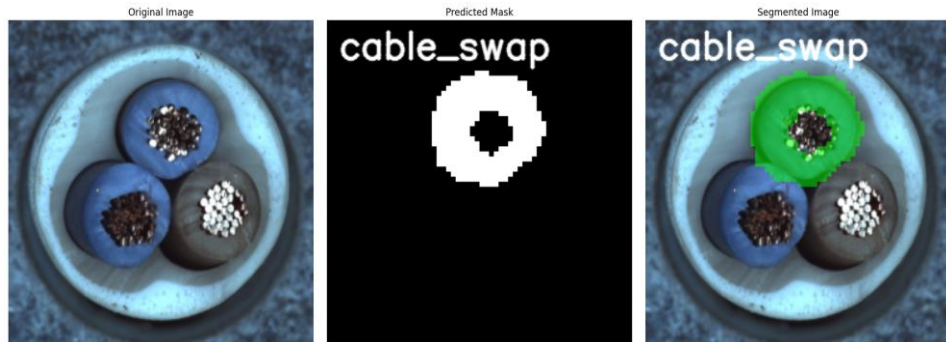


Figure 4. 84: Cable swap defect of Cable

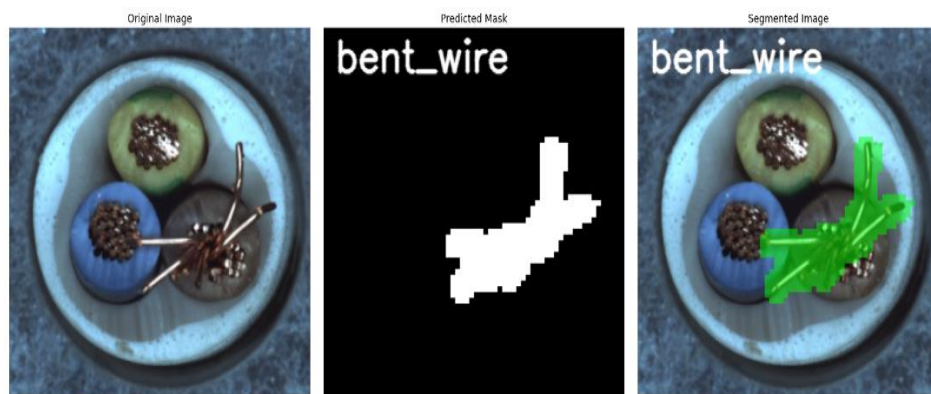


Figure 4. 85: Bent wire defect of Cable

4.4.2 Results obtained for Defect detection and segmentation for Grid using DeeplabV3+ model

This section describes learning curve from the trained model for grid, accuracy of the trained model for grid, predicted mask for the trained model for grid and Defected image with the predicted mask applied for grid.

4.4.2.1 Learning curve from the trained model for Grid using DeeplabV3+ model

The graph represents the training and validation performance of a machine learning model over 50 epochs. The Figure 4.86(a) illustrates the Training and Validation Loss, while the Figure 4. 86(b) shows the Training and Validation Accuracy.

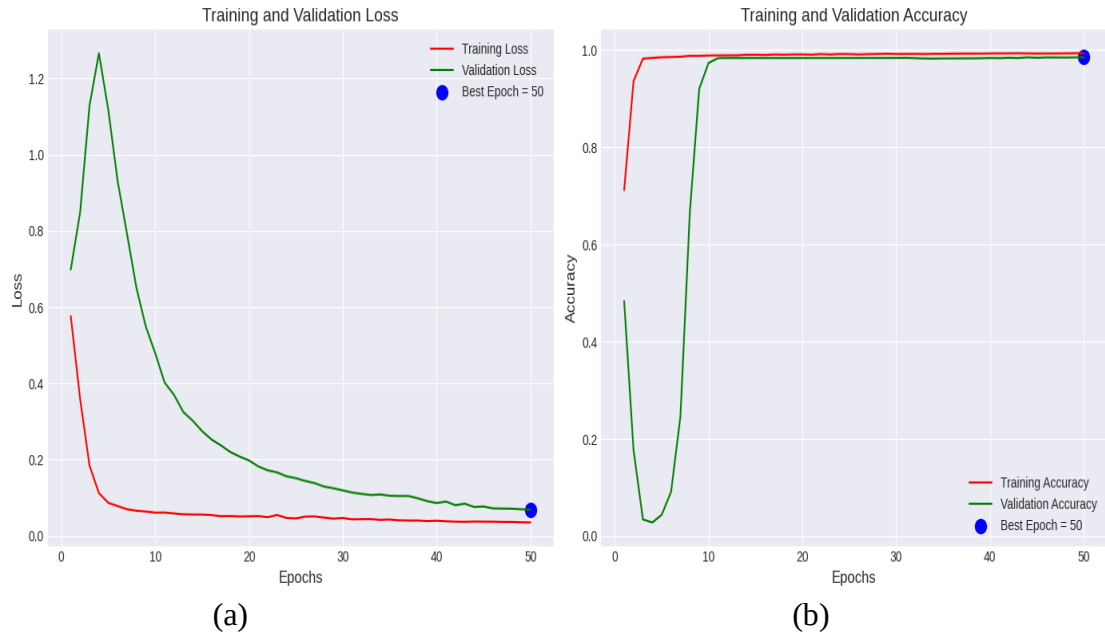


Figure 4.86: Learning curve for grid using DeeplabV3+ model

In the loss curves, we observe an initial sharp decline in both training and validation loss during the first 10 epochs, suggesting rapid learning and convergence. The training loss reaches a minimal value close to zero by the 10th epoch, indicating that the model is learning well from the training data. However, the validation loss, after an initial decline, experiences slight fluctuations, particularly during the first 10 epochs, before gradually stabilizing. By the 50th epoch, the validation loss converges to a low value, but the gap between the training and validation loss suggests some overfitting—the model fits the training data more closely than the validation data.

The accuracy curves depict rapid improvement in training accuracy, which quickly approaches 100% by the 10th epoch. The validation accuracy exhibits fluctuations during the initial epochs but stabilizes and aligns closely with the training accuracy after epoch 10, achieving nearly perfect accuracy by epoch 50. The blue marker on both graphs highlights the best epoch (epoch 50) for optimal validation performance. The stability in both accuracy curves and the convergence of the validation loss indicate that the model achieves strong generalization, although slight overfitting is suggested by the gap between the training and validation loss. Despite this, the overall

performance of the model is strong, with minimal validation loss and nearly perfect accuracy.

For future improvements, techniques such as regularization or early stopping could be employed to further reduce overfitting while maintaining the model's high accuracy. These learning curves serve as important indicators of the model's behavior and generalization capability.

4.4.2.2 Accuracy of the trained model for Grid

The evaluation results presented in Figure 4.87 depict the performance of the DeepLabV3+ model during the testing phase on the provided dataset. The model was evaluated over two batches (2/2), each step taking approximately 17 seconds, indicating moderate computational complexity per step. The model achieved a high accuracy of 0.9894, demonstrating that 98.94% of the pixels in the test set were correctly classified as either defective or non-defective. The reported loss of 0.0586 indicates minimal deviation between the predicted and actual segmentation labels, further affirming the model's capability to generalize well on unseen data. The detailed metrics [0.06320037692785263, 0.9883730411529541] provide additional granularity, with the first value representing the loss and the second value slightly refining the accuracy to 98.837%. These results indicate that the DeepLabV3+ model is highly effective for Defect detection in metallic surfaces, delivering precise and reliable segmentation performance, suitable for industrial defect detection applications.

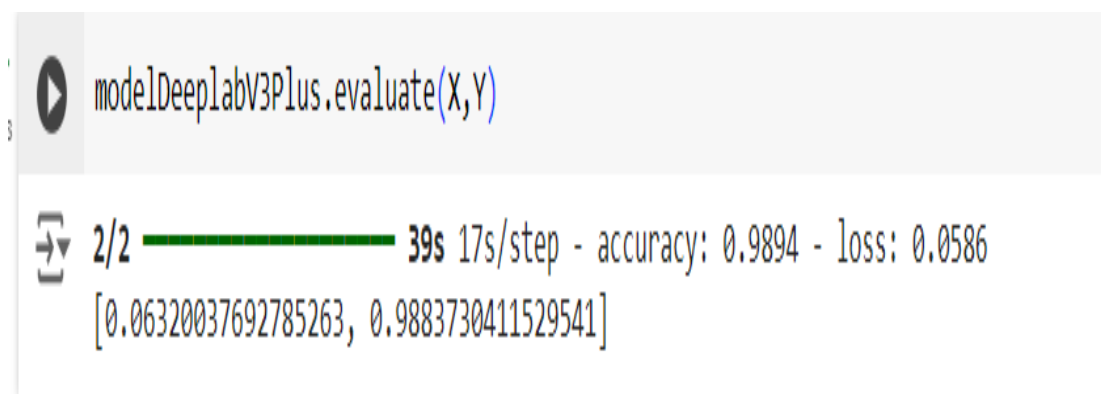
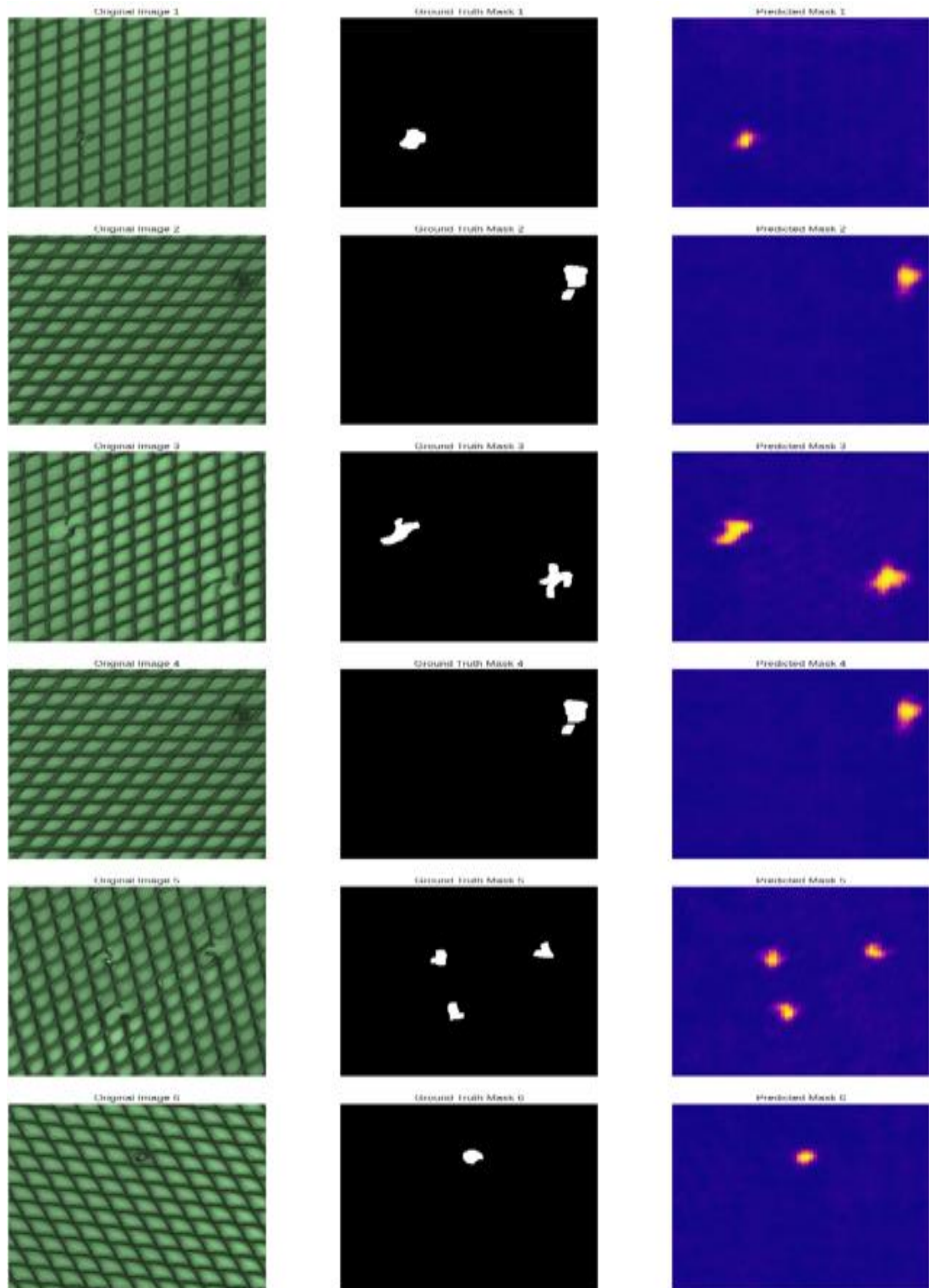


Figure 4. 87: Accuracy of the trained model for Grid

4.4.2.3 Predicted mask for the trained model for Grid

The Figure 4.88 shows the predicted mask of the different defected area of the Grid after training the model.



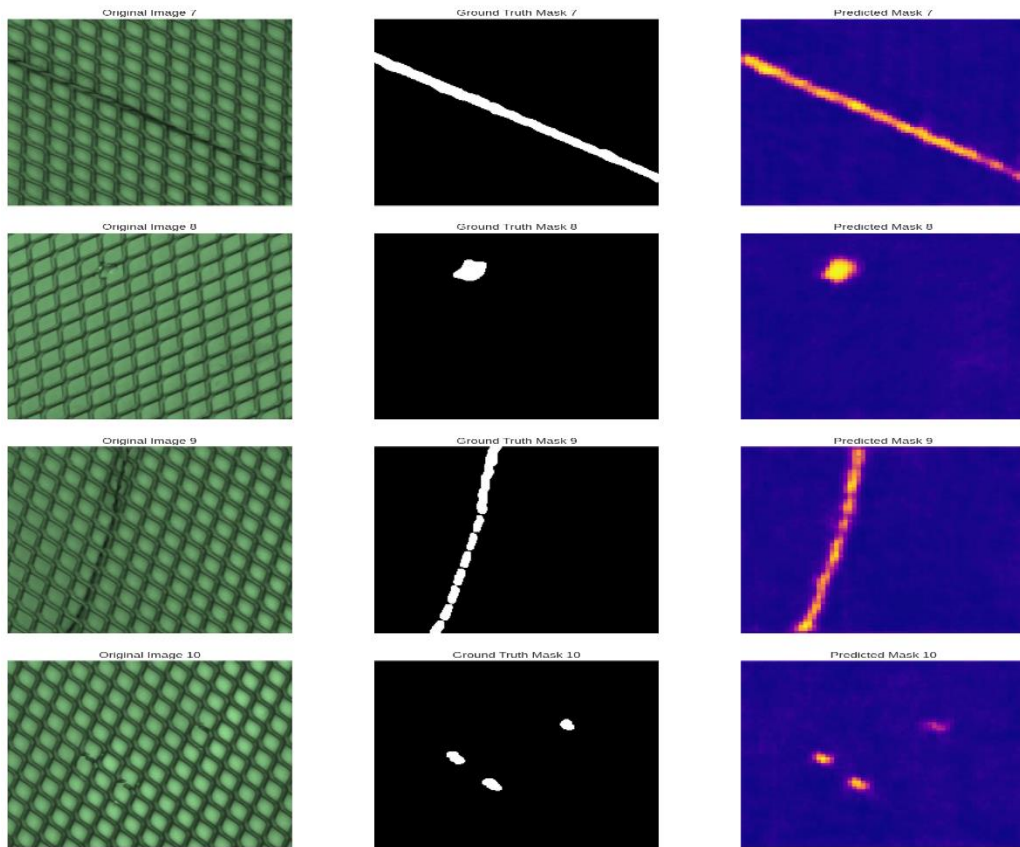


Figure 4.88: Predicted mask for the trained model for Grid

4.4.2.4 Defected image with the predicted mask applied for Grid

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of grid like bent, broken, glue defect are segmented fault in Figure 4.89-Figure 4.93.

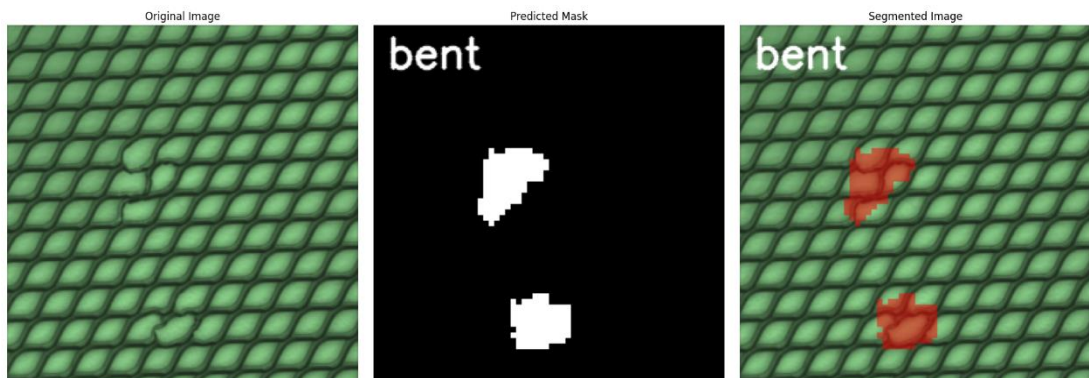


Figure 4.89: Bent defect of grid

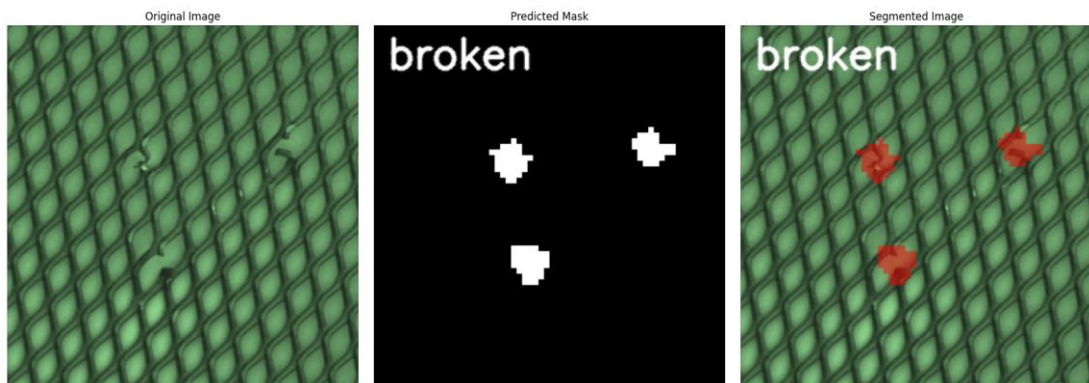


Figure 4.90: Broken defect of grid



Figure 4.91: Glue defect of grid

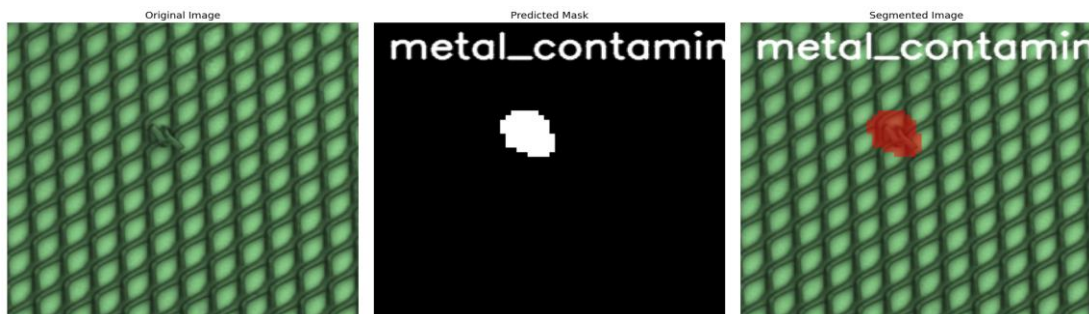


Figure 4.92: Metal contamination defect of grid

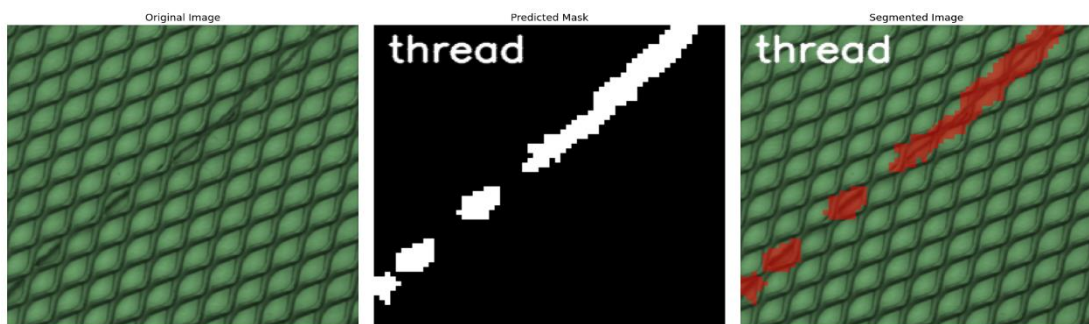


Figure 4.93: Thread defect of grid

4.4.3 Results obtained for Defect detection and segmentation for Metal nut using DeeplabV3+ model

This section describes learning curve from the trained model for Metal nut, accuracy of the trained model for Metal nut, predicted mask for the trained model for Metal nut and Defected image with the predicted mask applied for Metal nut.

4.4.3.1 Learning curve from the trained model for Metal nut using DeeplabV3+ model

The Figure 4. 94(a) presents the training and validation loss and Figure 4. 94(b) presents accuracy over 50 epochs for a machine learning model. The training loss decreases steadily, stabilizing at a low value, indicating consistent learning. However, the validation loss, while initially decreasing, shows fluctuations and a slower decline, suggesting variability in model performance on unseen data. The best epoch is marked at 50, where the validation loss is minimized. On the accuracy graph, the training accuracy quickly reaches above 90% and remains stable, while the validation accuracy starts lower, showing significant improvement up to epoch 15 (marked as the best epoch for validation accuracy) before fluctuating and gradually rising. The gap between training and validation accuracy indicates overfitting, as the model performs consistently better on the training set compared to the validation set. To address this, further regularization techniques or data augmentation might be needed to improve generalization and reduce the performance gap.

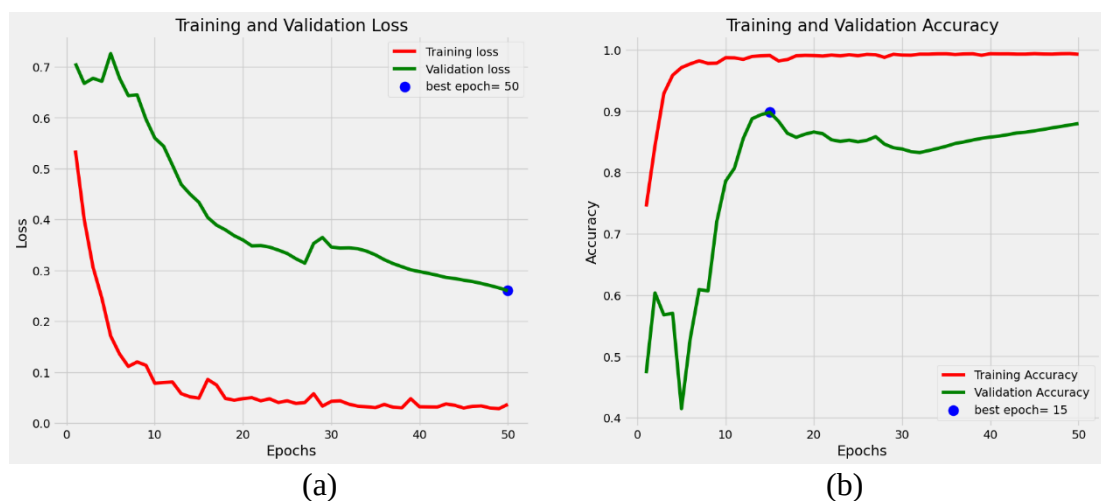


Figure 4. 94: Learning curve for metal nut using DeeplabV3+ model

4.4.3.2 Accuracy of the trained model for Metal nut

In this Figure 4.95, the evaluation output of a machine learning model, specifically the DeepLabV3+ architecture, designed for image segmentation tasks. The model was evaluated on a test dataset comprising three batches, with each batch requiring approximately 44 seconds and 11 milliseconds per step. The overall accuracy achieved by the model is 99.02%, indicating that the model is highly effective in correctly classifying the pixel-wise data within the segmentation task. Additionally, the model's loss is recorded at 0.0284, which is notably low, signifying minimal prediction error and well-converged learning during the evaluation phase. The output also provides further statistical metrics: a primary loss value of 0.03704182803630829 and a secondary accuracy score of 0.987283515990448. These metrics collectively demonstrate that the model has successfully learned the underlying patterns in the dataset, exhibiting both strong performance and generalization capabilities.

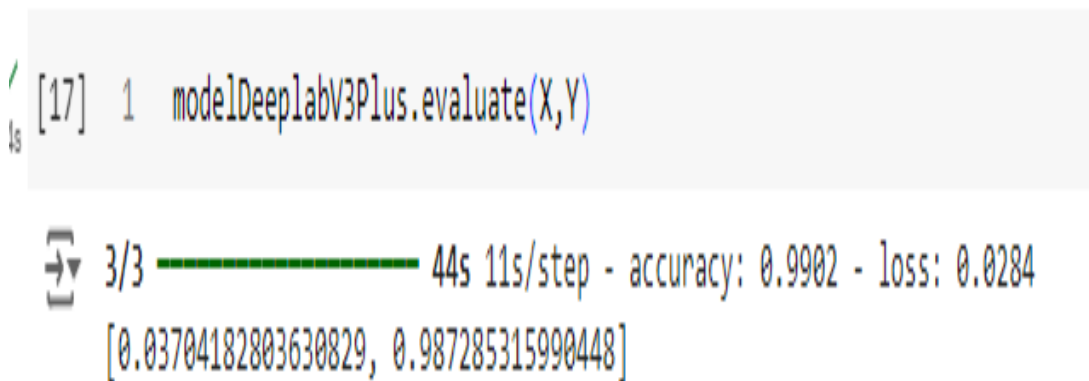


Figure 4. 95: Accuracy of the trained model for Metal nut

4.4.3.3 Predicted mask for the trained model for Metal nut

Figure 4.96 shows the predicted mask of the different defected area of the cable after training the model.

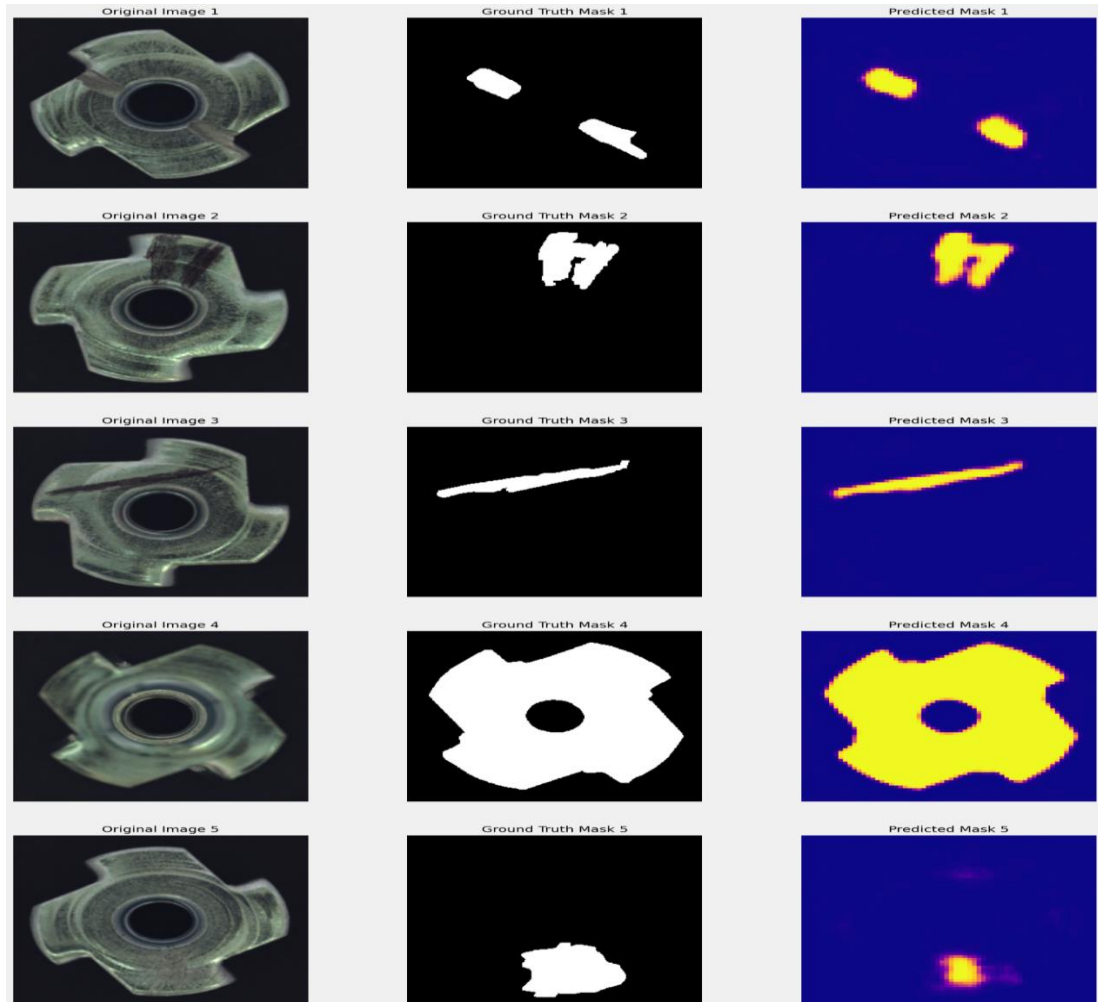


Figure 4. 96: Predicted mask for the trained model for Metal nut

4.4.3.4 Defected image with the predicted mask applied for Metal nut

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of Metal nut like bent, color, flip, scratch defect are segmented fault in Figure 4.97-Figure 4.99.

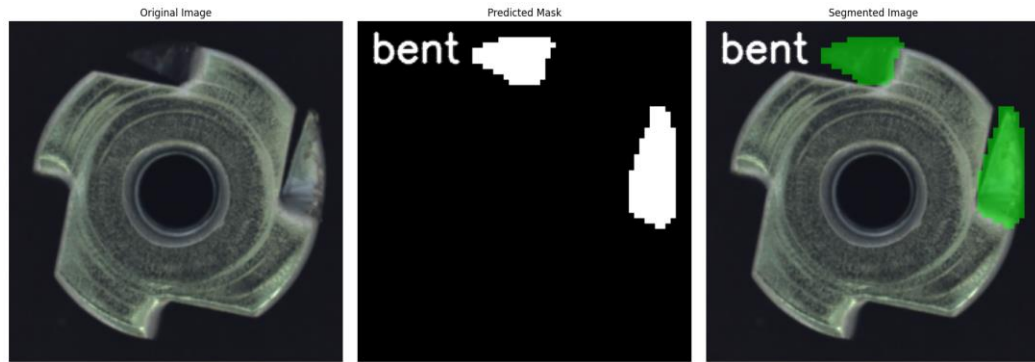


Figure 4.97: Bent defect of metal nut

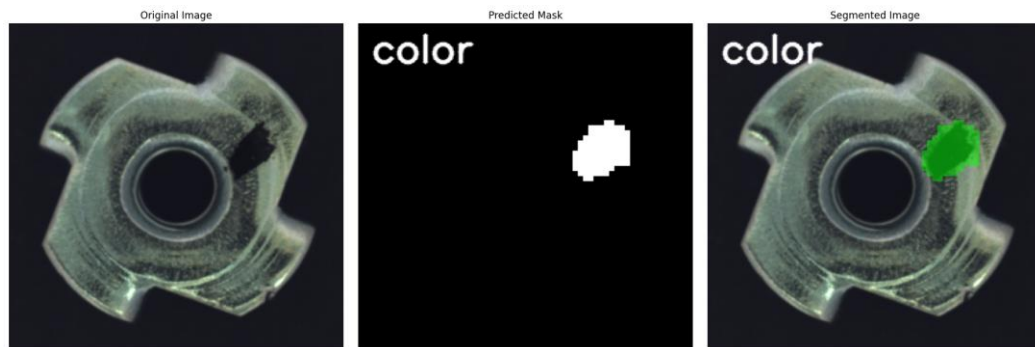


Figure 4.98: Color defect of metal nut

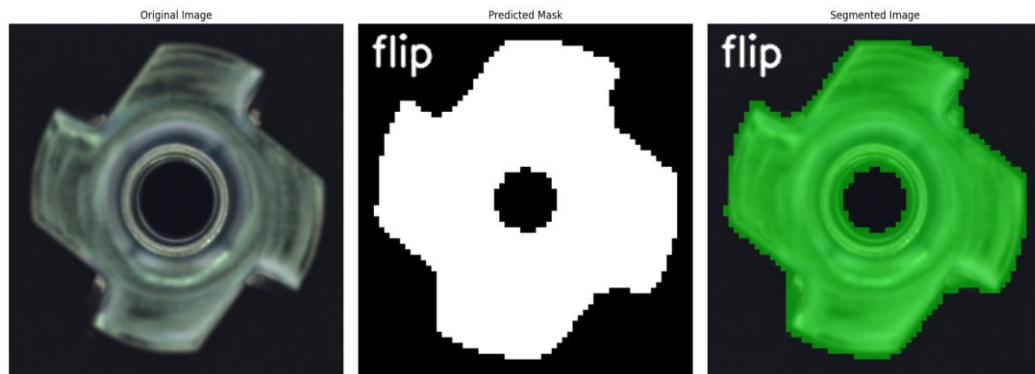


Figure 4.99: Flip defect of metal nut

4.4.4 Results obtained for Defect detection and segmentation for Screw using DeeplabV3+ model

This section describes learning curve from the trained model for screw, accuracy of the trained model for screw, predicted mask for the trained model for screw and Defected image with the predicted mask applied for screw.

4.4.4.1 Learning curve from the trained model for Screw using DeeplabV3+ model

The Figure 4.100(a) displays the training and validation loss and Figure 4.100(b) displays accuracy over 50 epochs for a machine learning model. The training loss

drops sharply within the initial epochs and then converges to near zero, indicating effective learning. The validation loss follows a similar pattern, decreasing significantly and stabilizing close to the training loss, showing that the model generalizes well to unseen data with minimal overfitting. The best epoch is marked as 38, where the validation loss reaches its minimum. In the accuracy graph, both training and validation accuracy rapidly climb to near-perfect levels (close to 100%) within the first few epochs and remain stable for the rest of the training period, demonstrating high performance and consistency. The minimal gap between training and validation accuracy suggests that the model is neither overfitting nor underfitting, indicating a well-optimized and balanced model.

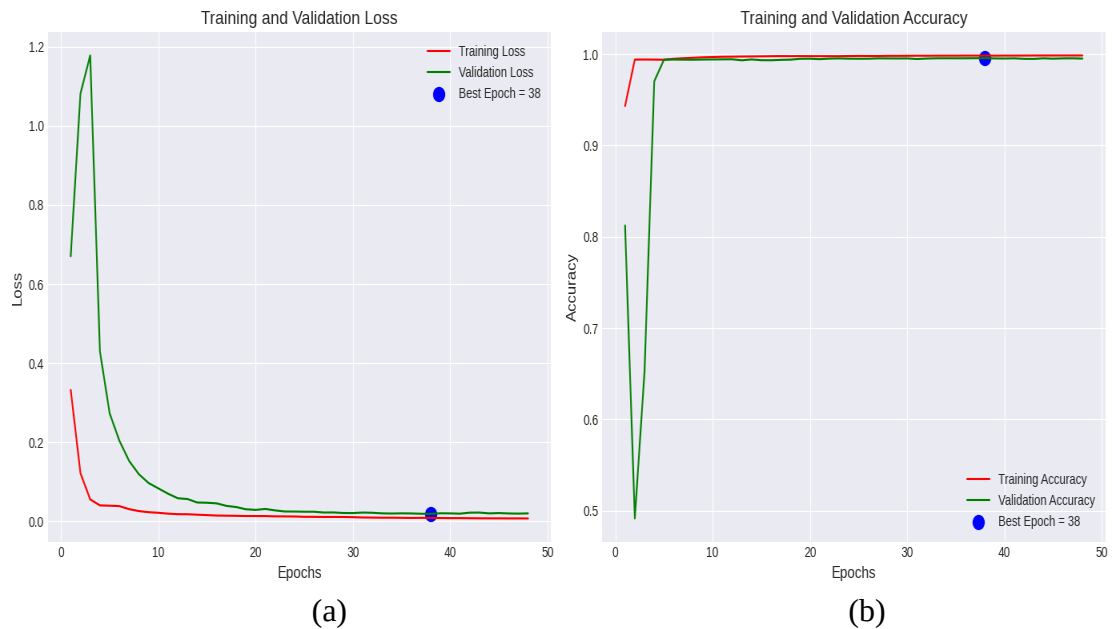


Figure 4.100: Learning curve for Screw using DeeplabV3+ model

4.4.4.2 Accuracy of the trained model for Screw

In this Figure 4.101, the evaluation of the DeepLabV3+ model on the test dataset is shown, demonstrating the model's performance across four batches. The model's accuracy during this evaluation is reported to be 99.78%, reflecting highly reliable performance in classifying the dataset accurately for the given task, which is likely image segmentation. The evaluation process took an average of 22 seconds per batch, with each step requiring approximately 6 seconds. The recorded loss is notably low at 0.0109, indicating minimal error in the model's predictions, suggesting strong

convergence and generalization from the training process. Additionally, detailed loss and accuracy metrics are provided, with a specific loss value of 0.010731774382293224 and a slightly higher secondary accuracy score of 99.77%. These results confirm that the model has achieved a robust understanding of the dataset and is capable of delivering near-perfect accuracy with minimal computational loss, further reinforcing its applicability for tasks requiring precise pixel-level classification.

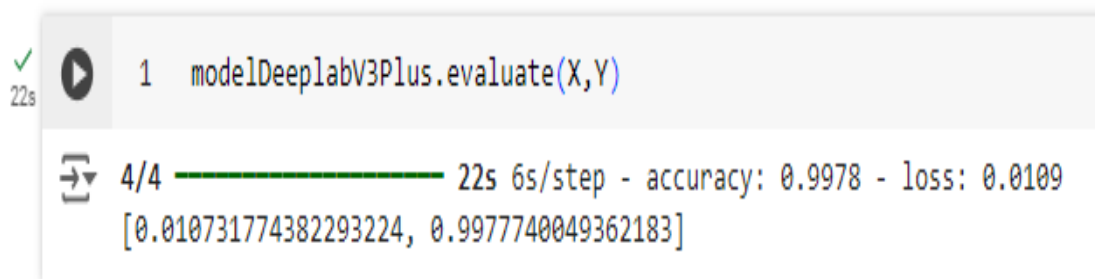
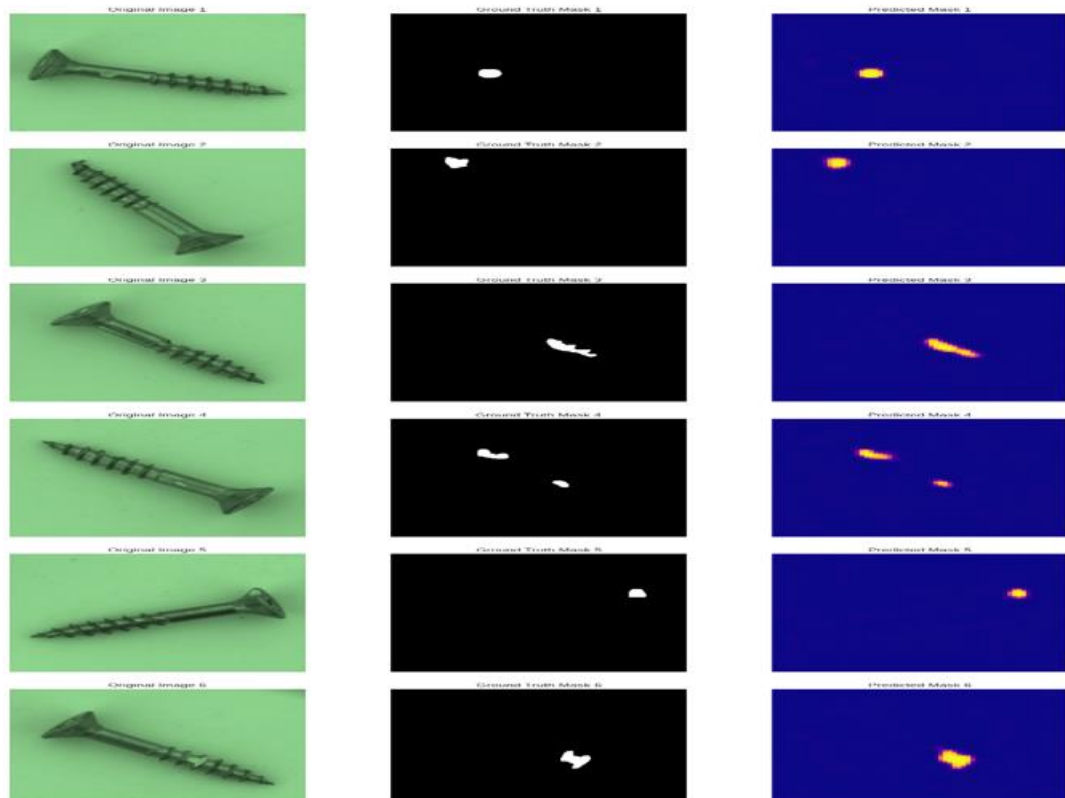


Figure 4.101: Accuracy of the trained model for Screw

4.4.4.3 Predicted mask for the trained model for Screw

Figure 4.102 shows the predicted mask of the different defected area of the cable after training the model.



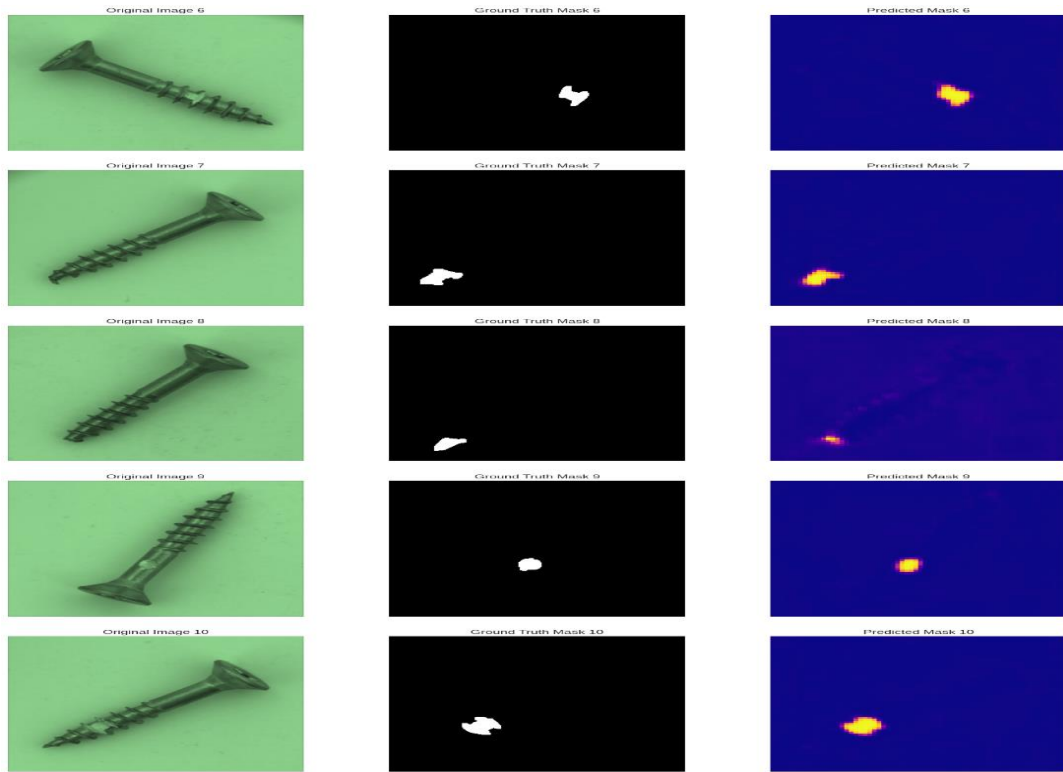


Figure 4.102: Predicted mask for the trained model for Screw

4.4.4.4 Defected image with the predicted mask applied for Screw

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of screw like manipulated front, scratch head, thread side defect are segmented fault in Figure 4.103-Figure 4.105.

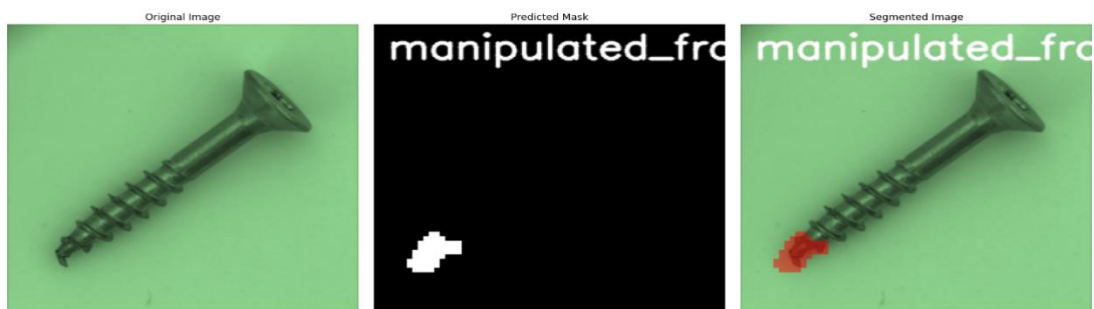


Figure 4.103: Manipulated front defect of screw

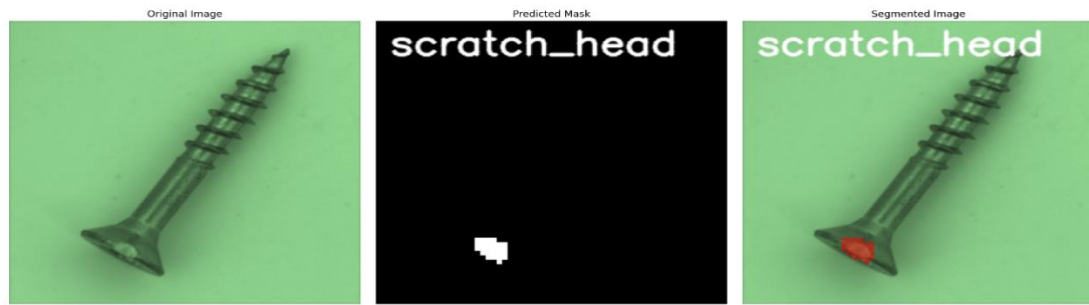


Figure 4.104: Scratch head defect of screw



Figure 4.105: Thread side defect of screw

4.4.5 Results obtained for Defect detection and segmentation for Transistor using DeeplabV3+ model

This section describes learning curve from the trained model for transistor, accuracy of the trained model for transistor, predicted mask for the trained model for transistor and Defected image with the predicted mask applied for transistor.

4.4.5.1 Learning curve from the trained model for transistor using DeeplabV3+ model

The Figure 4.106(a) illustrates the training and validation loss and Figure 4. 106(b) displays accuracy over 50 epochs for the model. The training loss decreases steadily, showing consistent convergence to a low value, while the validation loss starts high and fluctuates before gradually declining. This fluctuation in validation loss, compared to the stable training loss, indicates some variability in how the model generalizes to new data, although it eventually converges by epoch 49, marked as the best epoch. The accuracy graph shows that the training accuracy quickly reaches above 90% and remains stable throughout, while the validation accuracy starts lower and increases gradually, with significant improvement after epoch 10, eventually plateauing just below the training accuracy. The gap between training and validation accuracy

suggests some degree of overfitting, as the model performs better on the training data than on unseen data. To mitigate this, further tuning such as regularization or data augmentation might be necessary to improve the generalization performance of the model.

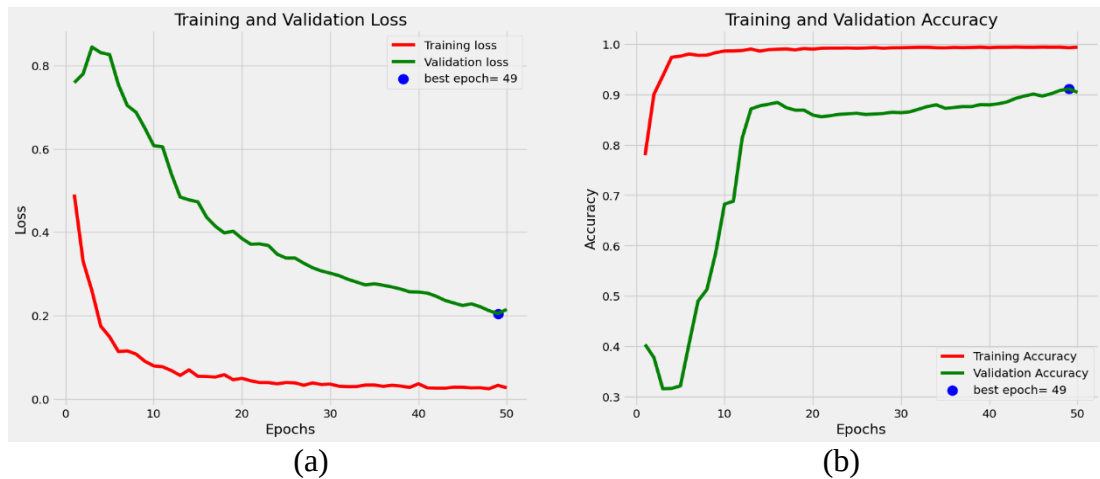


Figure 4.106: Learning curve for transistor using DeeplabV3+ model

4.4.5.2 Accuracy of the trained model for transistor

In this Figure 4.107, the evaluation output of the DeepLabV3+ model on a test dataset over two batches. The average time taken per step is around 5 seconds and 1 millisecond, indicating a relatively quick evaluation process. The accuracy achieved is 97.93%, signifying that the model is highly capable of predicting the correct outcomes with a strong performance level. However, the loss value is recorded at 0.0593, which, although still low, is higher compared to previous evaluations, suggesting a slight increase in prediction error. The specific breakdown of metrics includes a primary loss value of 0.0566466748714447 and a secondary accuracy score of 98.05%. This indicates that while the model continues to perform well, there is a minor but noticeable increase in loss, which may suggest areas for further fine-tuning or adjustment in model parameters to reduce error and enhance performance consistency.

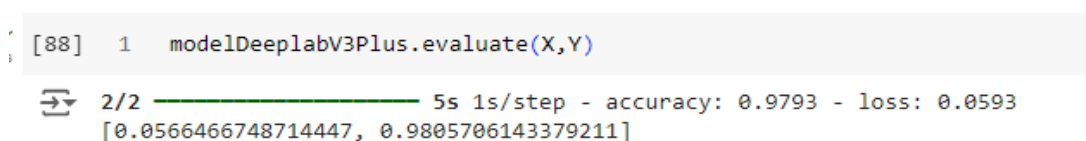


Figure 4.107: Accuracy of the trained model for transistor

4.4.5.3 Predicted mask for the trained model for transistor

Figure 4.108 shows the predicted mask of the different defected area of the cable after training the model

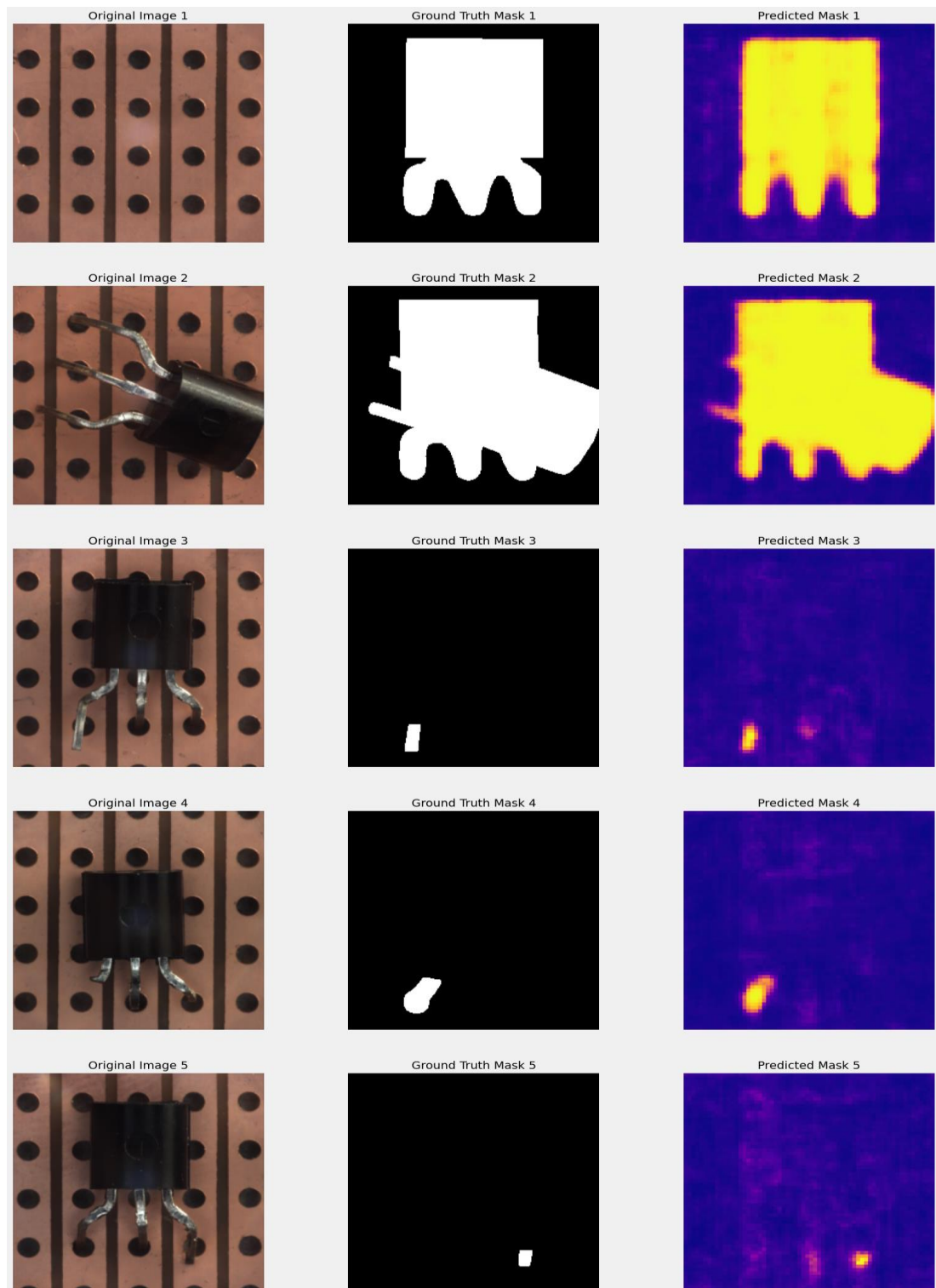


Figure 4.108: Predicted mask for the trained model for transistor

4.4.5.4 Defected image with the predicted mask applied for transistor

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of transistor like bent lead, cut lead, misplaced, damaged case defect are segmented fault in Figure 4.109-Figure 4.112.



Figure 4.109: Bent lead defect of transistor

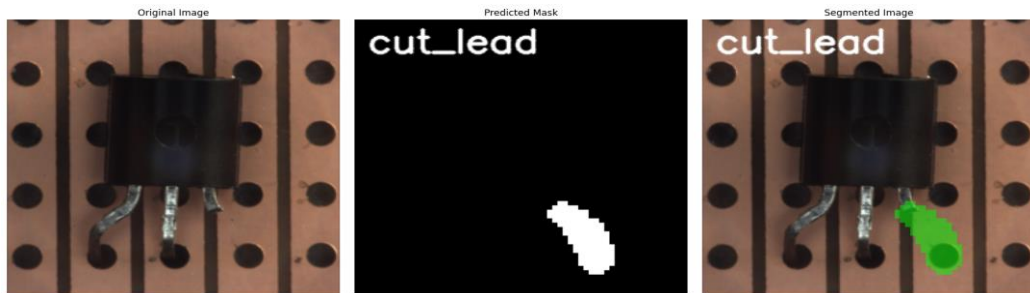


Figure 4.110: Cut lead defect of transistor



Figure 4.111: Damaged case defect of transistor

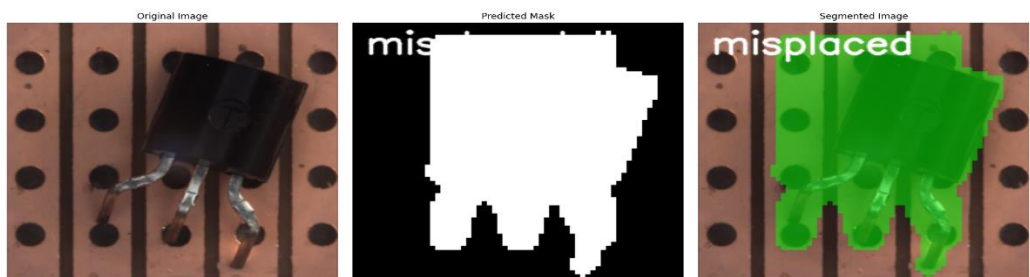


Figure 4.112: Misplaced defect of transistor

4.4.6 Results obtained for Defect detection and segmentation for Zipper using DeeplabV3+ model

This section describes learning curve from the trained model for Zipper, accuracy of the trained model for Zipper, predicted mask for the trained model for Zipper and Defected image with the predicted mask applied for Zipper.

4.4.6.1 Learning curve from the trained model for Zipper using DeeplabV3+ model

The Figure 4. 113(a) shows the training and validation loss and Figure 4. 113(b) shows accuracy over 50 epochs for the model. In the loss graph, both training and validation losses begin high, with the validation loss peaking initially before sharply decreasing and stabilizing around epoch 10. From there, both losses remain low and converge closely, indicating effective training with minimal overfitting. The best epoch is marked at 43, where the model achieves its lowest loss values. On the accuracy graph, both training and validation accuracy increase rapidly, reaching near 100% by epoch 5, and they remain consistent throughout the remaining epochs. The overlap of training and validation accuracy curves suggests that the model is generalizing well without signs of overfitting or underfitting. This indicates that the model has learned effectively and is performing optimally, achieving high accuracy and low loss on both training and validation datasets.

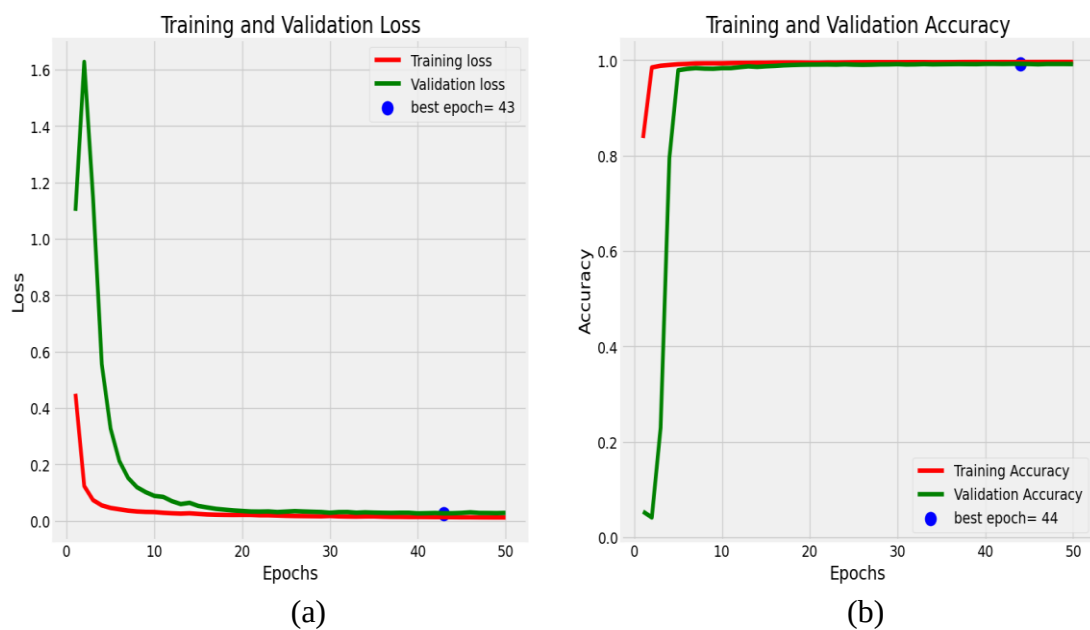


Figure 4.113: Learning curve for zipper using DeeplabV3+ model

4.4.6.2 Accuracy of the trained model for Zipper

In this Figure 4.114, the evaluation output of the DeepLabV3+ model over four batches are presented, showing that the model performed well with an accuracy of 99.64%. This high accuracy reflects the model's proficiency in accurately predicting the output for the given dataset. The evaluation time per step averaged 6 seconds, while the total evaluation duration was 40 seconds for all four batches. The loss value is reported at 0.0122, which indicates a minimal error rate, further reinforcing the model's strong learning and generalization abilities. Additionally, the specific loss value is 0.013492967002093792, and a secondary accuracy score of 99.60% is provided. These metrics confirm the DeepLabV3+ model's robust capability in segmenting images with high precision and low error, highlighting its effectiveness for tasks requiring detailed pixel-wise classification. This suggests that the model is well-optimized and capable of delivering consistent, accurate results in real-world applications.

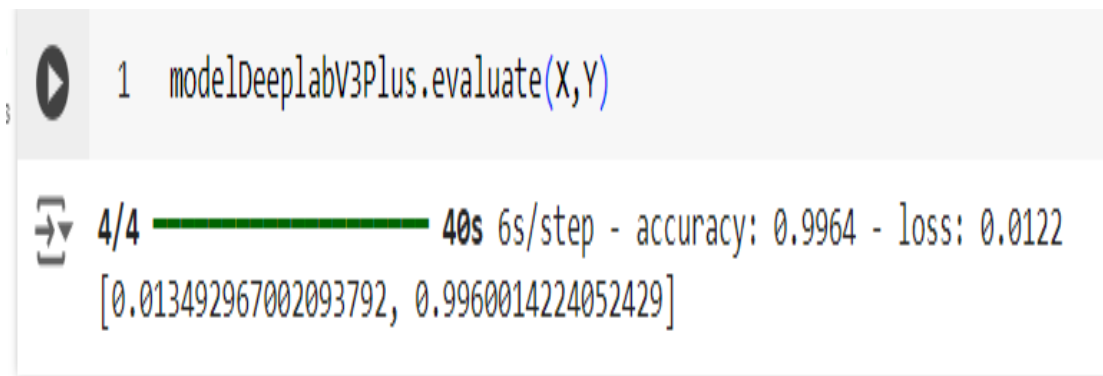


Figure 4.114: Accuracy of the trained model for Zipper

4.4.6.3 Predicted mask for the trained model for Zipper

Figure 4.115 shows the predicted mask of the different defected area of the cable after training the model

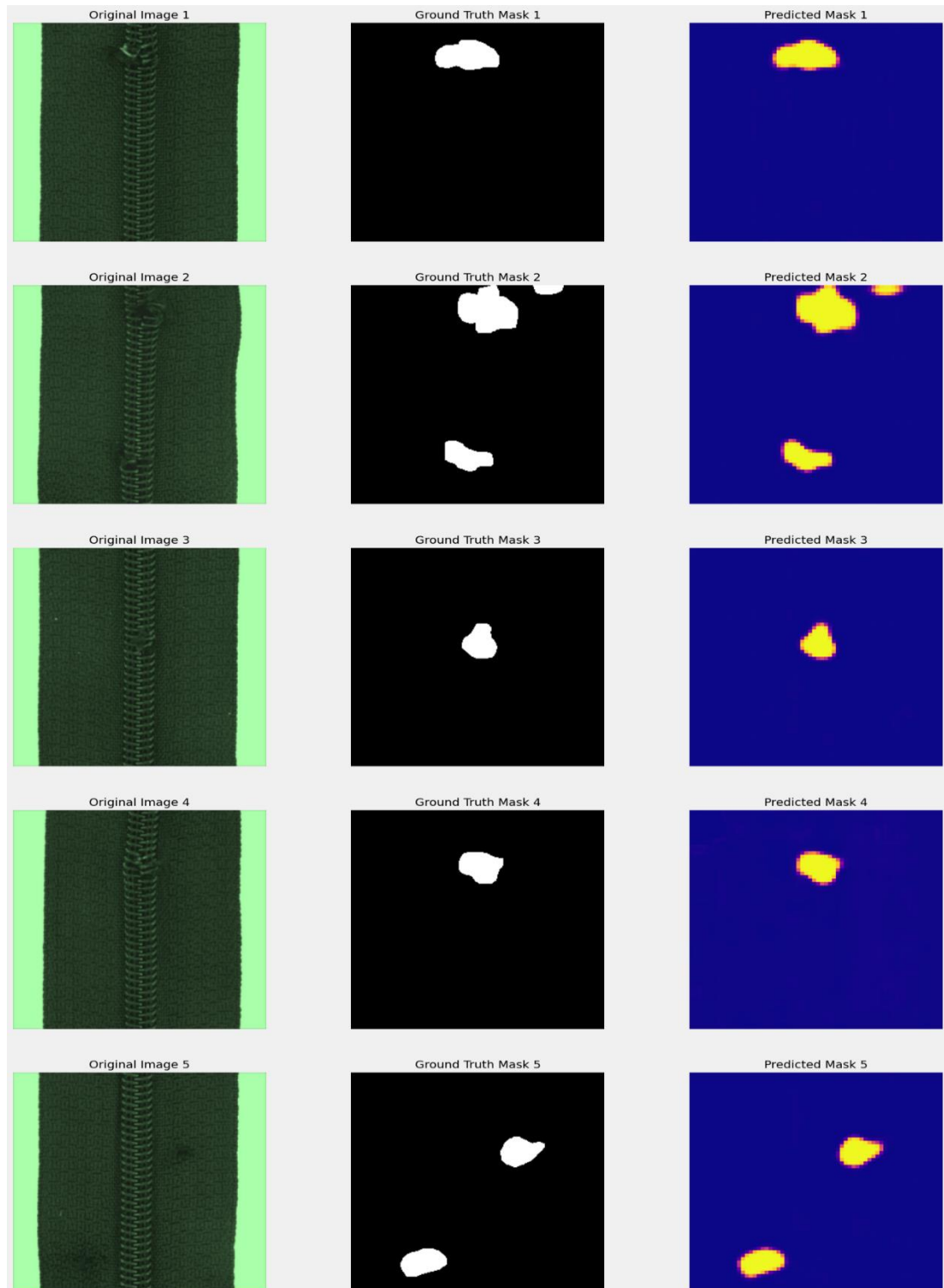


Figure 4.115: Predicted mask for the trained model for Zipper

4.4.6.4 Defected image with the predicted mask applied for Zipper

MVTec AD (MVTec Anomaly Detection) dataset contains 15 types of objects from which 6 metallic objects are picked for this study (Cable, Grid, Metal nut, Screw, Transistor, Zipper). In this section, different defects of Zipper like fabric border defect, fabric interior defect, rough defect, squeezed teeth defect, split teeth defect are segmented fault in Figure 4.116-Figure 4.120.



Figure 4.116: Fabric border defect of zipper



Figure 4.117: Fabric interior defect of zipper

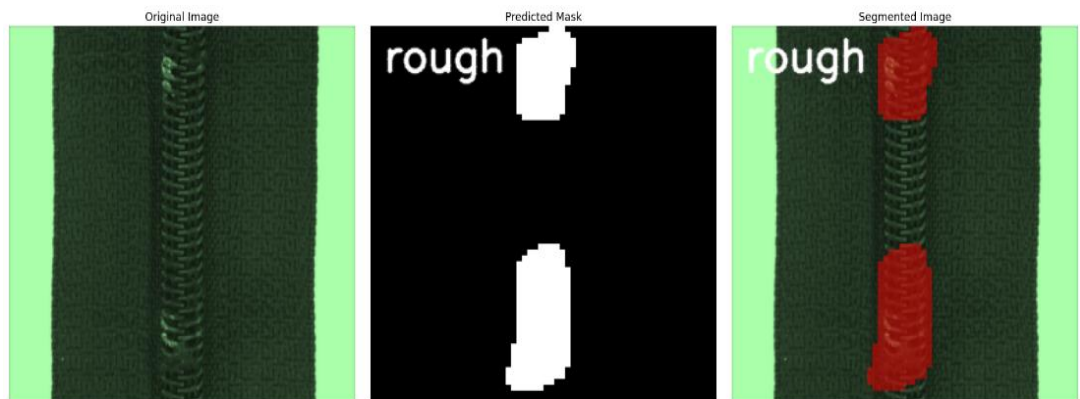


Figure 4.118: Rough defect of zipper

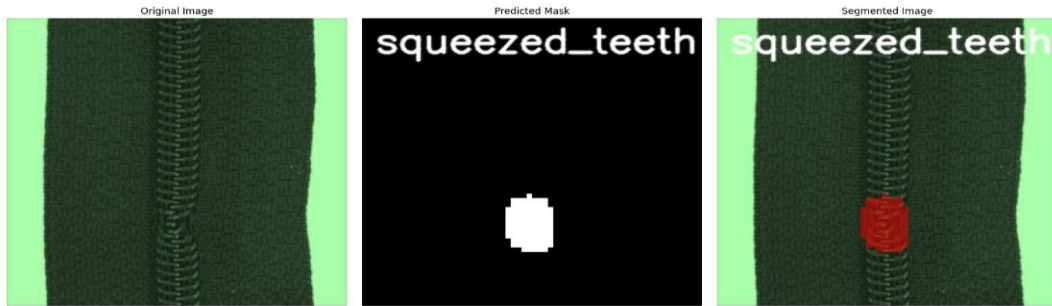


Figure 4.119: Squeezed teeth defect of zipper

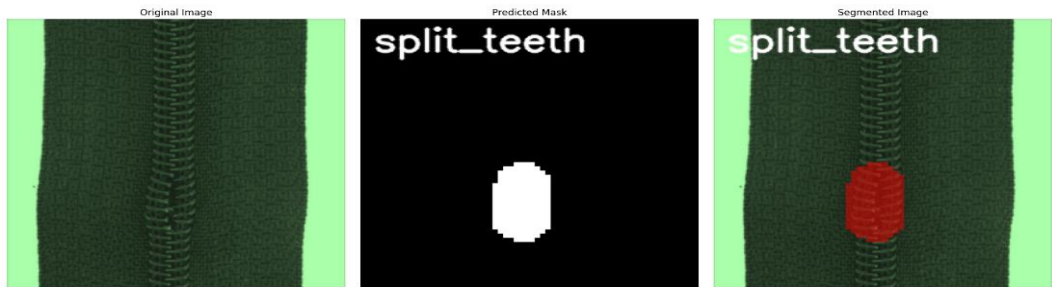


Figure 4.120: Split teeth defect of zipper

In this Table 4.3, the DeeplabV3+ model shows impressive accuracy across various object types, with consistently high performance.

Table 4. 3: Accuracy for different objects using DeeplabV3+ model

Algorithm	Object Type					
	Cable	Grid	Metal nut	Screw	Transistor	Zipper
DeeplabV3+ model	98.52%	98.94%	99.02%	99.78%	97.93%	99.64%

4.5 Conclusion

This thesis explored the application of machine learning techniques for Defect detection in metallic products, focusing on anomaly detection and defect segmentation. The central objective was to develop efficient, scalable models for detecting defects in metallic objects, which can significantly improve manufacturing quality control processes. The study utilized the MVTec Anomaly Detection dataset,

applying models like Mini Batch Dictionary Learning with Sparse Coder and Custom U-Net for segmentation.

The key findings from this research demonstrate the effectiveness of machine learning in improving defect detection accuracy, particularly when comparing unsupervised learning models. The Mini Batch Dictionary Learning approach with Sparse Coder provided significant improvements in anomaly detection across several metallic products, such as cables, grids, screws, and metal nuts. The model successfully detected a range of defects, including missing wires, bent components, scratches, and broken surfaces, achieving high precision in distinguishing between defective and non-defective images.

Additionally, the Custom U-Net model proved to be highly efficient in defect segmentation, especially when tasked with segmenting defects on metallic surfaces. The model achieved high accuracy rates across all tested categories, with performance metrics exceeding 90% in most cases. The segmentation results demonstrate that U-Net can effectively map and highlight defective areas, providing a more detailed analysis compared to traditional detection methods.

Furthermore, the results obtained using the DeepLabV3+ model also highlighted its capacity to handle complex segmentation tasks, reinforcing the potential of deep learning-based techniques in industrial applications. This model's ability to capture intricate details at the pixel level proves beneficial in identifying even the most subtle defects, further optimizing the defect detection pipeline.

Chapter 05

Conclusion

5.1 Introduction

The objective of this thesis was to explore the effectiveness of machine learning techniques in detecting faults and segmenting defects in metallic products. Given the critical role that product quality plays in industrial manufacturing, automating the detection of defects is an essential step toward improving operational efficiency and reducing costs. This chapter aims to summarize the key findings of the research, discuss their significance, and propose potential avenues for future research. The conclusions presented here are based on a comprehensive analysis of experimental results using various machine learning models, including Mini Batch Dictionary Learning with Sparse Coder, Custom U-Net, and DeepLabV3+.

5.2 Summary of findings

The research conducted in this study provides valuable insights into the use of machine learning algorithms for industrial Defect detection. The key findings are summarized as follows:

1. **Mini Batch Dictionary Learning with Sparse Coder:** This unsupervised approach for anomaly detection proved highly effective in identifying defects in a variety of metallic objects, including cables, grids, metal nuts, screws, and zippers. The model achieved high precision and recall rates, outperforming several benchmark algorithms in terms of detecting anomalies in metallic products.
2. **Custom U-Net for Defect Segmentation:** The Custom U-Net model demonstrated strong performance in segmenting defects in metallic surfaces. Its ability to generate accurate masks for defective areas was critical in identifying specific defect types, such as missing wires, bent components, and surface scratches. The model's accuracy ranged from 87% to 99%, depending on the object type, making it a robust solution for quality control in manufacturing.
3. **DeepLabV3+ for Fine-Grained Segmentation:** The implementation of DeepLabV3+ further enhanced defect segmentation, particularly in cases where

high-resolution precision was required. The model's ability to perform pixel-level segmentation allowed it to detect subtle defects that might have been missed by other models, providing a comprehensive solution for detecting and localizing defects.

The Table 5.1 summarizes the accuracy and effectiveness of three models—Mini Batch Dictionary Learning with Sparse Coder, Custom U-Net, and DeeplabV3+—for detecting defects in different objects using the MVTec AD dataset.

Table 5. 1: Summary of findings of this thesis

Model	Object Type	Dataset Used	Accuracy / AUROC	Notes
Mini Batch Dictionary Learning with Sparse Coder	Cable	MVTec AD	0.708 (AP)	Effective for detecting missing and cut cable defects.
	Grid	MVTec AD	0.976 (AP)	High precision for detecting bent and broken defects.
	Metal Nut	MVTec AD	0.814 (AP)	Strong performance in detecting surface scratches and bends.
	Screw	MVTec AD	0.747 (AP)	Effective detection of surface defects like scratches and thread misalignment.
	Transistor	MVTec AD	0.489 (AP)	Slightly lower precision in detecting complex defects such as misplaced components.
	Zipper	MVTec AD	0.987 (AP)	High accuracy for detecting various

				zipper defects, including broken teeth and fabric tears.
Custom U-Net	Cable	MVTec AD	96.92%	Strong segmentation of cable defects such as cut insulation.
	Grid	MVTec AD	98.33%	Accurate segmentation of defects like glue contamination and thread issues.
	Metal Nut	MVTec AD	87.92%	Good performance in defect segmentation, slightly lower for complex anomalies.
	Screw	MVTec AD	99.59%	Excellent segmentation accuracy for defects such as thread top irregularities.
	Transistor	MVTec AD	98.79%	High accuracy in detecting defects like bent leads and damaged cases.
	Zipper	MVTec AD	98.79%	High precision in defect segmentation, particularly for split and squeezed teeth defects.
	Cable	MVTec AD	98.52%	Excellent segmentation of cable defects, surpassing Mini Batch model in accuracy.

DeepLabV3+	Grid	MVTec AD	98.94%	Slightly higher accuracy in defect detection and segmentation for grid-related defects.
	Metal Nut	MVTec AD	99.02%	Best performance in the detection and segmentation of metal nut defects.
	Screw	MVTec AD	99.78%	Near-perfect segmentation of screw defects, including scratches and manipulated threads.
	Transistor	MVTec AD	97.93%	High accuracy in detecting transistor defects, slightly lower for fine defects.
	Zipper	MVTec AD	99.64%	Excellent segmentation of complex zipper defects, almost perfect performance.

Overall, the research confirmed that machine learning models, particularly those based on deep learning, are well-suited for industrial defect detection. These models not only improve the speed and accuracy of inspections but also minimize human error, reducing production costs and enhancing product quality. By automating these processes, industries can adopt more efficient, scalable, and sustainable manufacturing workflows.

5.3 Contribution of This Research

This research contributes significantly to the field of industrial defect detection and quality control by advancing the use of machine learning algorithms for detecting and segmenting surface defects in metallic products. The primary contributions of this work are outlined below:

1. Development of an Automated Defect Detection Framework: This study presents a comprehensive framework integrating multiple machine learning models—Mini Batch Dictionary Learning, Custom U-Net, and DeepLabV3+—to detect and segment defects in metallic products. Unlike traditional methods that require extensive manual inspection, the proposed system automates the entire defect detection process, reducing the reliance on human effort, minimizing errors, and enhancing inspection speed and accuracy.

2. Improvement in Defect Segmentation Accuracy: The integration of U-Net and DeepLabV3+ models for pixel-level segmentation significantly enhances defect localization on metallic surfaces. These models, specifically optimized for this study, offer high accuracy in defect segmentation, successfully identifying even subtle imperfections like scratches and bent components. This level of precision is crucial for industries that demand high-quality standards, such as automotive and electronics manufacturing.

3. Addressing Industrial Quality Control Needs through Anomaly Detection: By applying Mini Batch Dictionary Learning for anomaly detection, this research addresses the challenge of detecting unexpected or novel defects in metallic surfaces. The algorithm's capability to identify anomalies without labeled data is particularly valuable for industrial applications, where continuous production often introduces unforeseen defect types that supervised models may fail to recognize.

4. Advancement of Real-time Application Feasibility: The system architecture leverages GPU acceleration and efficient batch processing techniques, making it feasible for real-time deployment in industrial settings. This approach enables faster

inspection cycles and timely defect identification, aligning with the needs of high-throughput production environments. The study also highlights the adaptability of these models to varied industrial conditions, underscoring the system's robustness.

5. Reduction of Production Costs and Waste: By implementing automated, high-accuracy defect detection, this research contributes to lowering production costs by reducing the rejection rate of finished products. The system's precise defect identification minimizes unnecessary waste, promoting sustainable manufacturing practices. As industries seek to optimize resource utilization, this study provides a scalable, automated solution for maintaining high production quality with minimal waste.

6. Establishing a Foundation for Future Research: The findings and methodologies outlined in this thesis provide a foundation for further exploration into hybrid models and real-time analytics in defect detection. Future research could build on this work by integrating more advanced deep learning techniques, expanding the application to different types of defects, and exploring new performance metrics. This study thus opens up new avenues for advancing industrial anomaly detection and quality control systems.

Overall, this research offers a comprehensive and effective solution for industrial defect detection, blending machine learning techniques to address both the efficiency and accuracy challenges prevalent in traditional quality control methods.

5.4 Future research directions

The findings of this study lay the groundwork for future improvements and research in industrial Defect detection using machine learning. One potential avenue for future work includes exploring more advanced deep learning techniques, such as hybrid models that combine traditional ML with deep learning architectures. Moreover, integrating real-time data analytics systems into these models could enhance their applicability in real-world industrial settings, allowing for quicker detection and response times.

Another area worth exploring is the use of larger, more diverse datasets. The current study utilized a specific dataset focused on metallic products, but expanding the model to other industries could provide more generalized insights. Additionally, investigating the effect of different environmental factors on model performance could provide deeper insights into model robustness and real-time adaptability in dynamic manufacturing environments.

5.5 Better performance of these proposed models

The proposed model, which combines Mini Batch Dictionary Learning with Sparse Coder, Custom U-Net, and DeeplabV3+, offers several advantages over existing methods for defect detection in metallic products.

- 1. Improved Anomaly Detection:** The proposed model employs Mini Batch Dictionary Learning with Sparse Coder, which has been shown to be effective in detecting anomalies in high-dimensional data. This approach enables the model to identify defects that may not be apparent through traditional methods.
- 2. Enhanced Defect Segmentation:** The Custom U-Net architecture used in the proposed model is specifically designed for image segmentation tasks, allowing for precise identification and segmentation of defects in metallic products. This is particularly important in industrial settings where accurate defect detection and segmentation are crucial for quality control.
- 3. State-of-the-Art Performance:** The incorporation of DeeplabV3+ in the proposed model ensures state-of-the-art performance in image segmentation tasks. DeeplabV3+ has been shown to outperform other architectures in various benchmark datasets, making it an ideal choice for defect detection in metallic products.
- 4. Robustness to Noise and Variability:** The proposed model is robust to noise and variability in the data, which is common in industrial settings. The Mini Batch Dictionary Learning with Sparse Coder approach is effective in handling noisy data,

while the Custom U-Net and DeeplabV3+ architectures are designed to handle variability in the data.

5. Flexibility and Adaptability: The proposed model is flexible and adaptable to different types of metallic products and defects. The Custom U-Net architecture can be modified to accommodate different product types and defect characteristics, while the DeeplabV3+ architecture can be fine-tuned for specific tasks.

6. Efficient Computation: The proposed model is computationally efficient, allowing for fast processing of large datasets. This is particularly important in industrial settings where timely defect detection and segmentation are critical for quality control.

5.6 Comparison to Existing Methods

The proposed model offers several advantages over existing methods for defect detection in metallic products. For example:

1. Traditional Computer Vision Methods: Traditional computer vision methods, such as edge detection and thresholding, are limited in their ability to detect defects in metallic products. The proposed model offers improved anomaly detection and defect segmentation capabilities.

2. Deep Learning-Based Methods: Deep learning-based methods, such as convolutional neural networks (CNNs), have been shown to be effective in defect detection tasks. However, the proposed model offers improved performance and robustness to noise and variability.

3. Hybrid Methods: Hybrid methods, which combine traditional computer vision and deep learning-based approaches, have been proposed for defect detection tasks. However, the proposed model offers improved flexibility and adaptability to different product types and defect characteristics.

In conclusion, the proposed model offers several advantages over existing methods for defect detection in metallic products. Its improved anomaly detection, enhanced defect segmentation, state-of-the-art performance, robustness to noise and variability, flexibility, and adaptability make it an ideal choice for industrial settings.

5.7 Concluding remarks

The integration of machine learning into Defect detection in metallic products represents a promising solution for industries looking to optimize their manufacturing processes. By automating the defect detection and segmentation process, machine learning models reduce the reliance on human inspections, which can be error-prone and inefficient. This research demonstrates that leveraging models such as Mini Batch Dictionary Learning with Sparse Coder and deep learning-based segmentation architectures can significantly improve the accuracy and efficiency of defect detection, leading to better quality control, reduced costs, and enhanced operational efficiency in manufacturing industries. As technology continues to evolve, the integration of AI-driven solutions will undoubtedly play a critical role in the future of industrial automation.

References

- [1] P. M. Bhatt *et al.*, “Image-Based Surface Defect Detection Using Deep Learning: A Review,” *J Comput Inf Sci Eng*, vol. 21, no. 4, Aug. 2021, doi: 10.1115/1.4049535/1094064.
- [2] S. H. Park, K. H. Lee, J. S. Park, and Y. S. Shin, “Deep Learning-Based Defect Detection for Sustainable Smart Manufacturing,” *Sustainability (Switzerland)*, vol. 14, no. 5, Mar. 2022, doi: 10.3390/su14052697.
- [3] “Unveiling Precision: AIs Impact on Defect Detection in Manufacturing.” Accessed: Sep. 01, 2024. [Online]. Available: <https://praxie.com/ai-for-defect-detection-in-manufacturing/>
- [4] Y. LeCun, Y. Bengio, G. H.- nature, and undefined 2015, “Deep learning,” *nature.com* Y LeCun, Y Bengio, G Hintonnature, 2015•nature.com, 2015, doi: 10.1038/nature14539.
- [5] H. Shen, W. Du, W. Sun, Y. Xu, J. F.-A. Sciences, and undefined 2019, “Visual detection of surface defects based on self-feature comparison in robot 3-D printing,” *mdpi.com* H Shen, W Du, W Sun, Y Xu, J FuApplied Sciences, 2019•mdpi.com, doi: 10.3390/app10010235.
- [6] E. Westphal and H. Seitz, “A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks,” *Addit Manuf*, vol. 41, p. 101965, May 2021, doi: 10.1016/J.ADDMA.2021.101965.
- [7] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, “The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection,” *Int J Comput Vis*, vol. 129, no. 4, pp. 1038–1059, Apr. 2021, doi: 10.1007/s11263-020-01400-4.
- [8] H. Dong, K. Song, Y. He, J. Xu, Y. Yan, and Q. Meng, “PGA-Net: Pyramid Feature Fusion and Global Context Attention Network for Automated Surface Defect Detection,” *IEEE Trans Industr Inform*, vol. 16, no. 12, pp. 7448–7458, Dec. 2020, doi: 10.1109/TII.2019.2958826.

- [9] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTec AD-A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection." [Online]. Available: www.mi.imati.cnr.it/ettore/NanoTWICE/
- [10] E. Şanlıtürk, A. T. Tekin, and F. Çebi, "Defect Detection in Manufacturing via Machine Learning Algorithms," *Encyclopedia of Data Science and Machine Learning*, pp. 212–225, Jan. 2022, doi: 10.4018/978-1-7998-9220-5.CH013.
- [11] Y. Xu, J. Zhu, S. Xu, W. Liu, and X. Bian, "Automatic segmentation of metal surface defects with noise label," 2024, doi: 10.21203/rs.3.rs-4787816/v1.
- [12] L. Zhuang, H. Qi, and Z. Zhang, "The Automatic Rail Surface Multi-Flaw Identification Based on a Deep Learning Powered Framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12133–12143, Aug. 2022, doi: 10.1109/TITS.2021.3109949.
- [13] D. Lin, Y. Li, S. Prasad, T. L. Nwe, S. Dong, and Z. M. Oo, "CAM-guided Multi-Path Decoding U-Net with Triplet Feature Regularization for Defect Detection and Segmentation," *Knowl Based Syst*, vol. 228, Sep. 2021, doi: 10.1016/J.KNOSYS.2021.107272.
- [14] I. Aydin, E. Akin, and M. Karakose, "Defect classification based on deep features for railway tracks in sustainable transportation," *Appl Soft Comput*, vol. 111, p. 107706, Nov. 2021, doi: 10.1016/J.ASOC.2021.107706.
- [15] J. Leñena, F. A. Saiz, and I. Barandiaran, "Latent Diffusion Models to Enhance the Performance of Visual Defect Segmentation Networks in Steel Surface Inspection," *Sensors*, vol. 24, no. 18, Sep. 2024, doi: 10.3390/S24186016.
- [16] Z. Nie, J. Xu, and S. Zhang, "Analysis on DeepLabV3+ Performance for Automatic Steel Defects Detection," Apr. 2020, Accessed: Oct. 04, 2024. [Online]. Available: <https://arxiv.org/abs/2004.04822v2>
- [17] X. Zheng, S. Zheng, Y. Kong, and J. Chen, "Recent advances in surface defect inspection of industrial products using deep learning techniques," *The International Journal of Advanced Manufacturing Technology* 2021 113:1, vol. 113, no. 1, pp. 35–58, Jan. 2021, doi: 10.1007/S00170-021-06592-8.
- [18] C. C. Huang and X. P. Lin, "Study on Machine Learning Based Intelligent Defect Detection System," *MATEC Web of Conferences*, vol. 201, p. 01010, Sep. 2018, doi: 10.1051/MATECCONF/201820101010.

- [19] A. Birlutiu, A. Burlacu, M. Kadar, and D. Onita, "Defect detection in porcelain industry based on deep learning techniques," *Proceedings - 2017 19th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing, SYNASC 2017*, pp. 263–270, Nov. 2018, doi: 10.1109/SYNASC.2017.00049.
- [20] J. Yang, S. Li, Z. Wang, and G. Yang, "Real-Time Tiny Part Defect Detection System in Manufacturing Using Deep Learning," *IEEE Access*, vol. 7, pp. 89278–89291, 2019, doi: 10.1109/ACCESS.2019.2925561.
- [21] X. Tao, D. Zhang, W. Ma, X. Liu, and De Xu, "Automatic Metallic Surface Defect Detection and Recognition with Convolutional Neural Networks," *Applied Sciences 2018, Vol. 8, Page 1575*, vol. 8, no. 9, p. 1575, Sep. 2018, doi: 10.3390/APP8091575.
- [22] M. Farhan Khan *et al.*, "Real-time defect detection in 3D printing using machine learning," *Mater Today Proc*, vol. 42, pp. 521–528, Jan. 2021, doi: 10.1016/J.MATPR.2020.10.482.
- [23] E. Şanlıtürk, A. T. Tekin, and F. Çebi, "Defect Detection in Manufacturing via Machine Learning Algorithms," <https://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/978-1-7998-9220-5.ch013>, pp. 212–224, Jan. 1AD, doi: 10.4018/978-1-7998-9220-5.CH013.
- [24] P. Damacharla, M. V. Achuth Rao, J. Ringenberg, and A. Y. Javaid, "TLU-Net: A Deep Learning Approach for Automatic Steel Surface Defect Detection," *2021 International Conference on Applied Artificial Intelligence, ICAPAI 2021*, May 2021, doi: 10.1109/ICAPAI49758.2021.9462060.
- [25] S. Jain, G. Seth, A. Paruthi, U. Soni, and G. Kumar, "Synthetic data augmentation for surface defect detection and classification using deep learning," *J Intell Manuf*, vol. 33, no. 4, pp. 1007–1020, Apr. 2022, doi: 10.1007/S10845-020-01710-X/FIGURES/13.
- [26] W. Zhao, F. Chen, H. Huang, D. Li, and W. Cheng, "A New Steel Defect Detection Algorithm Based on Deep Learning," *Comput Intell Neurosci*, vol. 2021, no. 1, p. 5592878, Jan. 2021, doi: 10.1155/2021/5592878.

- [27] D. Tabernik, S. Šela, J. Skvarč, and D. Skočaj, “Segmentation-based deep-learning approach for surface-defect detection,” *J Intell Manuf*, vol. 31, no. 3, pp. 759–776, Mar. 2020, doi: 10.1007/S10845-019-01476-X/FIGURES/17.
- [28] S. A. Singh and K. A. Desai, “Automated surface defect detection framework using machine vision and convolutional neural networks,” *J Intell Manuf*, vol. 34, no. 4, pp. 1995–2011, Apr. 2023, doi: 10.1007/S10845-021-01878-W/TABLES/7.
- [29] K. Demir, M. Ay, M. Cavas, and F. Demir, “Automated steel surface defect detection and classification using a new deep learning-based approach,” *Neural Comput Appl*, vol. 35, no. 11, pp. 8389–8406, Apr. 2023, doi: 10.1007/S00521-022-08112-5/TABLES/9.
- [30] Y. Huang, J. Jing, and Z. Wang, “Fabric Defect Segmentation Method Based on Deep Learning,” *IEEE Trans Instrum Meas*, vol. 70, 2021, doi: 10.1109/TIM.2020.3047190.
- [31] S.-T. Liong, Y. S. Gan, Y.-C. Huang, C.-A. Yuan, and H.-C. Chang, “Automatic Defect Segmentation on Leather with Deep Learning,” Mar. 2019, Accessed: Sep. 12, 2024. [Online]. Available: <https://arxiv.org/abs/1903.12139v1>
- [32] Y. Wu, Y. Qin, Y. Qian, F. Guo, Z. Wang, and L. Jia, “Hybrid deep learning architecture for rail surface segmentation and surface defect detection,” *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 2, pp. 227–244, Feb. 2022, doi: 10.1111/MICE.12710.
- [33] D. M. Tsai, S. K. S. Fan, and Y. H. Chou, “Auto-Annotated Deep Segmentation for Surface Defect Detection,” *IEEE Trans Instrum Meas*, vol. 70, 2021, doi: 10.1109/TIM.2021.3087826.
- [34] I. A. Okaro, S. Jayasinghe, C. Sutcliffe, K. Black, P. Paoletti, and P. L. Green, “Automatic Defect detection for laser powder-bed fusion using semi-supervised machine learning,” *Addit Manuf*, vol. 27, pp. 42–53, May 2019, doi: 10.1016/J.ADDMA.2019.01.006.
- [35] Z. Kang, C. Catal, and B. Tekinerdogan, “Machine learning applications in production lines: A systematic literature review,” *Comput Ind Eng*, vol. 149, p. 106773, Nov. 2020, doi: 10.1016/J.CIE.2020.106773.

- [36] E. Westphal and H. Seitz, "A machine learning method for defect detection and visualization in selective laser sintering based on convolutional neural networks," *Addit Manuf*, vol. 41, p. 101965, May 2021, doi: 10.1016/J.ADDMA.2021.101965.
- [37] X. Tao, D. Zhang, W. Ma, X. Liu, and De Xu, "Automatic Metallic Surface Defect Detection and Recognition with Convolutional Neural Networks," *Applied Sciences* 2018, Vol. 8, Page 1575, vol. 8, no. 9, p. 1575, Sep. 2018, doi: 10.3390/APP8091575.
- [38] Y. J. Park, S. K. S. Fan, and C. Y. Hsu, "A Review on Defect detection and Process Diagnostics in Industrial Processes," *Processes* 2020, Vol. 8, Page 1123, vol. 8, no. 9, p. 1123, Sep. 2020, doi: 10.3390/PR8091123.
- [39] J. P. Yun, W. C. Shin, G. Koo, M. S. Kim, C. Lee, and S. J. Lee, "Automated defect inspection system for metal surfaces based on deep learning and data augmentation," *J Manuf Syst*, vol. 55, pp. 317–324, Apr. 2020, doi: 10.1016/J.JMSY.2020.03.009.
- [40] J. Yang, S. Li, Z. Wang, H. Dong, J. Wang, and S. Tang, "Using Deep Learning to Detect Defects in Manufacturing: A Comprehensive Survey and Current Challenges," *Materials* 2020, Vol. 13, Page 5755, vol. 13, no. 24, p. 5755, Dec. 2020, doi: 10.3390/MA13245755.
- [41] R. Rai, M. K. Tiwari, D. Ivanov, and A. Dolgui, "Machine learning in manufacturing and industry 4.0 applications," *Int J Prod Res*, vol. 59, no. 16, pp. 4773–4778, 2021, doi: 10.1080/00207543.2021.1956675.
- [42] J. Yang, S. Li, Z. Wang, and G. Yang, "Real-Time Tiny Part Defect Detection System in Manufacturing Using Deep Learning," *IEEE Access*, vol. 7, pp. 89278–89291, 2019, doi: 10.1109/ACCESS.2019.2925561.
- [43] M. Farhan Khan *et al.*, "Real-time defect detection in 3D printing using machine learning," *Mater Today Proc*, vol. 42, pp. 521–528, 2020, doi: 10.1016/J.MATPR.2020.10.482.
- [44] R. Zhu, B. Xin, N. Deng, and M. Fan, "Semantic Segmentation Using DeepLabv3+ Model for Fabric Defect Detection," *Wuhan University Journal of Natural Sciences*, vol. 27, no. 6, pp. 539–549, Dec. 2022, doi: 10.1051/WUJNS/2022276539.

- [45] E. Güçlü, I. Aydin, and E. Akin, “Improved Deeplabv3+ Based Steel Surface Defect Segmentation,” *2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems, ICETIS 2024*, pp. 609–613, 2024, doi: 10.1109/ICETIS61505.2024.10459666.
- [46] Q. Zhou, Z. Situ, S. Teng, H. Liu, W. Chen, and G. Chen, “Automatic sewer defect detection and severity quantification based on pixel-level semantic segmentation,” *Tunnelling and Underground Space Technology*, vol. 123, p. 104403, May 2022, doi: 10.1016/J.TUST.2022.104403.
- [47] Y. Zhu, S. Liu, X. Wu, L. Gao, and Y. Xu, “Multi-class segmentation of navel orange surface defects based on improved DeepLabv3+,” *Journal of Agricultural Engineering*, vol. 55, no. 2, Feb. 2024, doi: 10.4081/JAE.2024.1564.
- [48] R. Dong and S. Xiao, “Surface defect detection of rollers based on improved DeepLabv3+ and migration learning,” <https://doi.org/10.1117/12.3039512>, vol. 13259, pp. 845–850, Sep. 2024, doi: 10.1117/12.3039512.
- [49] D. A. Yudin, V. Adeshkin, A. V. Dolzhenko, A. Polyakov, and A. E. Naumov, “Roof Defect Segmentation on Aerial Images Using Neural Networks,” *Studies in Computational Intelligence*, vol. 925 SCI, pp. 175–183, 2021, doi: 10.1007/978-3-030-60577-3_20.
- [50] “What is Dictionary Learning.” Accessed: Sep. 13, 2024. [Online]. Available: <https://www.activeloop.ai/resources/glossary/dictionary-learning/>
- [51] “Beginners Guide to VGG16 Implementation in Keras | Built In.” Accessed: Sep. 13, 2024. [Online]. Available: <https://builtin.com/machine-learning/vgg16>
- [52] “Everything you need to know about VGG16 | by Great Learning | Medium.” Accessed: Sep. 13, 2024. [Online]. Available: <https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918>
- [53] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 4510–4520, Jan. 2018, doi: 10.1109/CVPR.2018.00474.

- [54] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11211 LNCS, pp. 833–851, Feb. 2018, doi: 10.1007/978-3-030-01234-2_49.
- [55] “DeepLabv3 & DeepLabv3+ The Ultimate PyTorch Guide.” Accessed: Oct. 04, 2024. [Online]. Available: <https://learnopencv.com/deeplabv3-ultimate-guide/>
- [56] “DeepLabv3+ | CloudFactory Computer Vision Wiki.” Accessed: Oct. 04, 2024. [Online]. Available: <https://wiki.cloudfactory.com/docs/mp-wiki/model-architectures/deeplabv3>